Master Thesis

Academic Year 2009-2010

Master Thesis in Intelligent Systems

Proyecto Fin de Máster en Sistemas Inteligentes

WEEV: A Multidisciplinary Approach to Educational Game Development

Author:

Eugenio J. Marchiori

Advisors:

Baltasar Fernández Manjón, Ph.D.

Pablo Moreno-Ger, Ph.D.

Madrid, June 2010

Master of Research in Computer Science | *Máster en Investigación en Informática*

School of Computer Science | *Facultad de Informática*

Complutense University at Madrid | *Universidad Complutense de Madrid*

This page is intentionally left blank

El/la abajo firmante, matriculado/a en el Master en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: "*WEEV: A Multidisciplinary Approach to Educational Game Development*", realizado durante el curso académico 2009-2010 bajo la dirección de Dr. Baltasar Fernández Manjón y Dr. Pablo Moreno-Ger en el Departamento de Ingeniería del Software e Inteligencia Artificial, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Eugenio Jorge Marchiori

Madrid, 21 de junio de 2010

ACKNOWLEDGMENTS

After more than a year of work, thousands of lines of code, endless reading and re-readings of books and papers, etc. the most important results in this thesis still come from the collaboration with, support from and advice of other people.

I would like to start thanking my thesis advisors, Baltasar Fernández Manjón and Pablo Moreno-Ger, for their support, their trust and mostly their endless complains, corrections and "suggestions". Their experience, and the time they invested, was paramount to the completion of this work and all related achievements.

The work of the rest of the "<e-Adventure> team" and the <e-UCM> research group cannot go unacknowledged. The role of Javier and Ángel, not only in regard to the <e-Adventure> project, but also in proofing and correcting my ideas, suggesting new solutions and making all the working hours more enjoyable, stimulating and challenging deserves special thanks.

I would also like to acknowledge the time invested by all those that tested the alpha versions of the system, and their patience with the bugs and problems that were still to be found.

Moreover, I would like to make a general "thank you" to all the people in my life: friends, family and Carmen, that were always more than willing to support the chaotic life and absurd times my work imposed on them and keep me company along the way.

Finally, if you believe your effort is not correctly acknowledge here, please forgive my omission and accept my most sincere "Thank you".

ABSTRACT

This thesis presents the WEEV methodology: a comprehensive approach for the development of educational video games. WEEV is based on a multidisciplinary approach that includes narrative theory, educational theory, video game theory; and visual language theory, as well as other aspects of computer science.

The methodology presented in this work is introduced as a way to include educators and domain-experts in the educational video game development process, without the need for them to understand scripting languages or complex logic. To achieve these goals it uses a high-level visual representation of the story flow that provides an abstraction of low-level logic and system interactions.

A working implementation of the system was developed following software engineering design principles and was evaluated by users in two different settings. These evaluations have served a double purpose: the system was improved based on user feedback and the potential of the WEEV approach to educational video game development was validated.

RESUMEN

Este trabajo presenta la metodología WEEV: un enfoque global para el desarrollo de videojuegos educativos. WEEV se basa en un enfoque multidisciplinario que incluye teoría narrativa, teoría educativa, teoría de videojuegos, y teoría lenguajes visuales, así como otros aspectos de informática.

La metodología presentada en este trabajo se presenta como una manera de incluir a los educadores y expertos en el dominio en el proceso de desarrollo de videojuegos educativos, sin la necesidad de que entiendan lenguajes de *scripting* o lógica compleja. Para alcanzar estos objetivos se utiliza una representación visual de alto nivel del flujo de la historia que proporciona una abstracción de la lógica de bajo nivel y de las interacciones con el sistema.

Una implementación funcional del sistema ha sido desarrollada siguiendo los principios de ingeniería de diseño de software y fue evaluada por usuarios en dos escenarios diferentes. Estas evaluaciones han servido un doble propósito: se mejoró el sistema en base a la retroalimentación de los usuarios y se validó el potencial del enfoque WEEV para el desarrollo de videojuegos educativos.

KEYWORDS

- Educational video games
- e-Learning
- Domain-Specific Visual Languages
- Authoring tool
- <e-Adventure>
- Education
- Game creation
- User-Centric Computing
- Human-Computer Interaction
- Serious games

PALABRAS CLAVE

- Video juegos educativos
- e-Learning
- Lenguajes visuales de dominio específico
- Herramienta de autoría
- <e-Adventure>
- Educación
- Creación de video juegos
- Computación centrada en usuario
- Interacción persona-ordenador
- Video juegos educativos

TABLE OF CONTENTS

INDEX OF FIGURES

INDEX OF TABLES

This page is intentionally left blank

CHAPTER I. INTRODUCTION

This chapter provides an introduction to the contents of this thesis. This includes the reasons why we believe this work is relevant and its main goals.

1. The case for educational game creation tools

The relevance and acceptance of educational video games (a.k.a. serious games) as a complement to traditional learning and as support tools in e-learning environments has been growing in recent years (Van Eck, 2006). This growth is reflected in the increasing number of educational games available in ever more diverse contexts and the increase in their scope and complexity.

However, the investment required to develop games has increased to reach multi-million dollar budgets[1] while the investment usually available for educational video games is still very limited. The game industry is poorly suited to fill this gap, as it is ever more focused on the development of big-budget blockbuster games thus limiting investment on small developments or developments for niche audiences. At the same time, instructors, teachers and domain experts usually lack the resources and knowledge needed to develop a game from scratch on their own.

We argue that game development tools and platforms are the adequate instruments to fill this gap. Existing game development tools originally designed for entertainment have been used in educational environments, however, the lack of some features (i.e. SCORM compliance) were considered important drawbacks (Academic ADL Co-Lab, 2004).

Specific platforms for educational game development can achieve cost reduction, provide simplified tools for a non-technical audience and conform with establish educational standards and specifications. Advances have been made in the cost reduction in educational game development through the use of specific tools (P. Moreno-Ger*, et al.*, 2005). At the same time, progress has been achieved in the simplification of tools to better fit the computer knowledge of domain-experts (J. Torrente*, et al.*, 2008). Furthermore, games created with such tools have been proven to be well received by learners and to improve educational outcomes (Pablo Moreno-Ger*, et al.*, 2010). Tools such as the <e-Adventure> platform provide different features that are specifically suited for education such as evaluation mechanisms and integration with Learning Management Systems (LMS) (Del Blanco*, et al.*, 2009).

Still, all these approaches are usually created as simplifications of existing systems, reusing the same metaphors and require that the developer acquires new knowledge and adapts his/her expectations to the expressive possibilities of the platform. Such limitations usually imply that the users avoid including complex in-game behaviors, and that game development requires time and special effort in learning and mastering the platform.

---

[1] *http://www.develop-online.net/news/33625/Study-Average-dev-cost-as-high-as-28m (retrieved on June 21st 2010)*

2. The limits of the current approach to educational game creation

Current approaches to educational game creation are mainly based on programming and as such require at least partial knowledge of the field. Some tools reduce the need for programming knowledge by the introduction of a metaphor identified as object-based or content-centric game creation. These tools are most successful in independent or amateur game creation, as the case of *Adventure Game Studio* shows. These tools have been complemented with educational aspects, including features both inside and outside the content-centric metaphor, with great success (Pablo Moreno-Ger, *et al.*, 2008).

However, these tools still present difficulties such as a steep learning curve. Another problem identified in these systems is the difficulty to re-use patterns, given that the game logic is scattered around several places and cannot be used independently from the rest of the game. We believe that an approach to video game creation that addresses these issues can present benefit to the end user that go beyond those of current systems.

Even though extending current metaphors can solve some of these issues, such a solution would either add complexity or reduce the scope covered by the tools. We propose a different approach based on changing the underlying metaphor used to create video games. The new metaphor we propose takes advantage of the work done in other fields and is based on the idea that games can be understood as a narrative, at least from the player's perspective.

3. A narrative metaphor for educational game creation

Based on different interpretations of narrative theory, a general metaphor for game creation can be extracted. Different educators have proposed the creation of games to teach story writing to their students (Robertson*, et al.*, 2004) or the use of the game development process as inspiration for composition curricula (Robison, 2008). Different authors have argued both for and against the consideration of video games as a narrative medium (Juul, 2005; Ryan, 2006) and different proposals have been made as to how a game could be described as a story (Dickey, 2006; Lindley, 2005).

All these different perceptions, proposals and experiences can be used to infer how a game could be created based on narrative concepts. The central idea behind such an approach is that different narratives are instantiated in each game run and for each player, and all the potential narratives that can be instantiated must be represented. We consider that the representation of the story is the main challenge in the development of a narrative metaphor.

Certain characteristics were identified as requirements for the representation of the story: it has to be explicit, it must be possible to modify it, it must represent the flow of the game and, last but not least, it must allow the forking of stories and paths, central to generating variations and challenges in a game. The representation presented in this thesis, possessing all these characteristics, uses a Domain-Specific Visual Language (DSVL) to describe a state diagram representing the flow of the game's story.

As shown by studies in narrative theory, the creation of the story *per se* is just part of the process (Dickey, 2006). The world where the story develops, the characters in the story and other game elements must be identified prior to the actual writing. These features of story writing are also reflected within the narrative metaphor.

4. WEEV: Writing Environment for Educational Video games

The work presented in this thesis is implemented, both from a theoretical and practical perspectives, into a system named WEEV (Writing Environment for Educational Video games). The system allows for the creation of educational video games using a narrative metaphor. The actual implementation has been created applying usability criteria to make it easier to understand, learn and use by users with limited knowledge of programming and game development techniques.

The WEEV methodology is contextualized within a theoretical framework. The narrative metaphor is based on theoretical proposals, using narrative theory, and complemented by studies of the video game development process and general story writing methods.

The implementation of the methodology into a system is supported by the use of Domain-Specific Visual Language (DSVL) and usability studies. This implementation was created in Java. The latest working version (beta 0.1 as of June 21st 2010) is available for download[2].

The goals of the system presented as the core work of this thesis are:

- Reduce the programming skills required to create games
- Create an explicit representation of the game story, which is easier to understand and can be used as documentation
- Introduce educational features within the same metaphor as the rest of the development process

[2] *http://sourceforge.net/projects/e-adventure/files/WEEV/WEEV_beta0.1.zip (retrieved June 21st 2010)*

5. Goals

This thesis as a whole presents a comprehensive approach to educational game development, covering everything from the game story representation, definition of the game interaction and the production of a playable game. This approach is strongly based on narrative theory and, in particular, its relationship with video games and video game development. The approach was implemented in the open-source WEEV system as part of the <e-Adventure> game development platform. Besides, the system and the approach have been subjected to formative evaluations with actual users. As such, the main goals of this work are:

- Establish a theoretical framework that supports the use of explicit narrative metaphors in game development
- Infer a systematic approach for educational video game development based on high-level narrative concepts
- Propose a concrete implementation of the narrative approach to educational video game development, in the way of the WEEV system
- Define a DSVL capable of expressing the inherit complexity of video games in a simple and straightforward way
- Perform formative evaluations of the WEEV system to establish its usability, usefulness and potential

CHAPTER II. THEORETICAL BACKGROUND

This chapter presents different theoretical positions and proposals regarding educational video games, narrative theory and their relationship. It also includes other relevant research to the development of WEEV (e.g. Visual Domain-Specific Language theory). All these ideas and proposals are studied in the context of this thesis and their applicability to the development of the new approach.

1. Educational video games

Different perceptions, definitions and types of educational video games exist. Studying the state of the art allows for some important facts about serious games to be established, as well as to provide a background about the use that such games are destined to have.

Educational video games have become a recognized complement to traditional educational approaches. Gee (2003) argues that games can be used to enhance learning. Amory *et al.* (1999) argue that computer games could provide a superior mechanism to entice learners to acquire knowledge through intrinsic motivation. Computer games can engage students in the learning environment, supporting contemporary educational practices (Amory, 2006). Also known as serious games, these games have grown both in number and complexity, as shown by the growth in investment in educational games (Wexler, *et al.*, 2008).

Two main groups can be identified in this field: Specific or custom developments and COTS (Commercial-Of-The-Shelf) games. The latter group is out of the scope of this work, but includes games such as the *SimCity*™ or *Civilization*™ sagas (Figure 1) being used in classroom environments to teach history, management principles or other subjects (Adams, 1998; Frye, *et al.*, 1996).

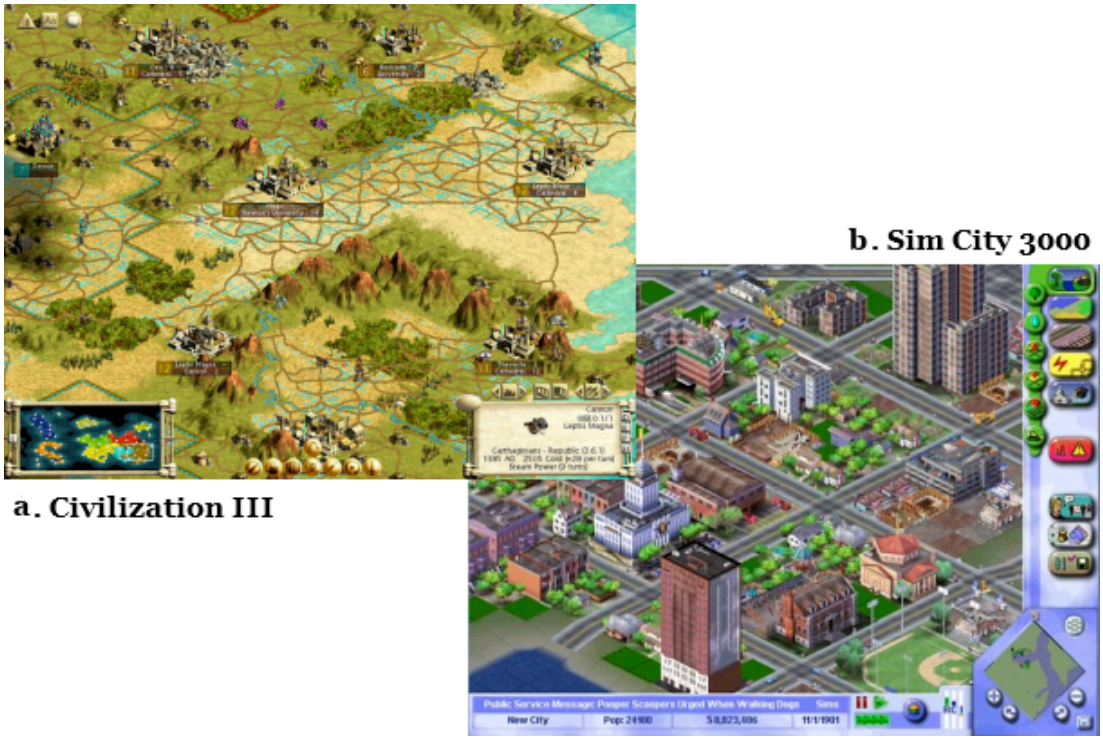


Figure 1 Screen-shots of (a) *Civilization III*™ and (b) *Sim City 3000*™, two COTS games used as educational tools

Specific developments include any game that was developed with the specific aim of being educational. This covers a broad range of games, including both games developed inside the game industry to teach basic subjects (e.g. math games) and those developed by specific interest groups (e.g. local governments) or educators.

Serious games developed inside the industry are mostly dependant on proprietary and/or expensive technologies and usually distributed as closed environments (i.e. black-box approach). These games are designed to fit general needs and allow little or no customization, thus limiting their target audience and reusability. Still, some games of this group have achieved widespread success, such as that of the games for the *Leapster*™ platform[3]. The *Leapster*™ platform includes different portable game consoles, targeted at different age groups, which run (usually franchised) educational games. Most specific developments created by interest groups, usually developed using technologies such as *Adobe Flash*™, mostly follow the same distribution approach.

Specific developments can, however, benefit from being open (i.e. white-box approach), easy to develop and created or adapted to fit the needs of teachers, professors, trainers or other educators that will use them. This can be achieved by the use of different platforms created specifically for educational games. Some of the benefits that can be achieved this way are:

- Better adaptation to the needs of particular groups
- More control of the subject by the educator
- Cost reduction (achieved by content reuse and open or free platforms)
- Longer amortization periods (games can be adapted every year)
- Improve instructional value by the reutilization of proven approaches

Besides, game development platforms that provide these characteristics can have alterative uses. Teachers have experimented by using games to teach story writing, not by the games themselves, but through the game creation process (Robison, 2008).

Educational video game design

This work presents a new approach to the design of educational video game that is either complementary, opposed or an extension of other approaches. Many different aspects must be considered in the design of educational games. These aspects range from the kind of motivation that will guide the students in the learning process to the importance given by learners to the different features of games. This study of the current state of the art in educational video game design establishes the most important aspects that influence good design practices regarding the narrative and story.

Kiili (2005) argues that the story is fundamental in game design, as it helps the extremely important task of immersing and engaging the player. Besides, Kiili argues that the story can be educative, providing non-interactive story events that tell the player important things about the subject matter.

According to Dickey (2006) the narrative (i.e. story) in adventure games "supports problem solving in complex, multimodal environments both by providing motivation and by serving as a cognitive framework". The literary techniques used by adventure games to achieve this are plot hooks (i.e.

[3] *http://shop.leapfrog.com/leapfrog/*

unanswered questions, usually delivered in cut-scenes) and emotional proximity (i.e. the empathy and identification the player feels towards the game character). Dickey (2005) identifies three elements of interactive design: *setting*; *roles and characters*; and *"hooks" that afford actions and feedback to the player*. The *setting* plays a role supporting the narrative, providing a sense of immersion and defining the *gamespace*. The *roles and characters* are defined depending on the game genre, in adventure games the player is usually the main character, and the characters and dialogues help establish a sense of immersion or *telepresence* in the game play environment. Importance must be placed in the creation of compelling characters, as the role-playing afforded by the player creating a bond with the main character has a positive influence in the learning experience. Finally, *actions, feedback and affordances* define what the player can do, as well as the victory and loss conditions. Rules must be consistent with the character and importance must be placed in "hooks" that keep the players playing the game.

Malone (1981) argues that fantasies (defined as "mental images of things not present to the senses or within the actual experience of the person involved") can make instructional environments more interesting and more educational. The notion of fantasy introduced by Malone is highly dependant on the story of the game, which introduces and enhances the fantasy setting. The author makes distinctions between different fantasies: extrinsic fantasies, where the fantasy depends on the use of the skill but not vice versa; and intrinsic fantasies, where the fantasy depends on the skill, but the skill also depends on the fantasy. In the latter kind, problems are presented with elements of the fantasy and so is the feedback. According to Malone, intrinsic fantasies are both more interesting and more instructional than extrinsic fantasies. At the same time, intrinsic fantasies have a greater dependence on the story, as both problems and feedback must be created within the same fantasy setting.

Studies conducted by Amory *et al.* (1999) show that students exposed to educational video games considered that the storyline, together with the graphics and sound, where the most important aspects of the games. Through further analysis Amory (2006) argues that "educational games should be designed as *narrative spaces* where *story* and *plot* (rhetoric acts) allow players to actively construct their own meaning/understanding through the use of plot devices that can include *back-story* and *cut scenes*". The author identifies five "narrative interfaces" in educational game development: *Narrative spaces*, *Challenges*, *Story*, *Plot* and *Back-story*.

All these analysis support our interest in improving the stories in educational video games to achieve better educational outcomes. These ideas are central to the development of the WEEV approach, based on making the story an integral part of the game creation process. This is achieved by the introduction of narrative theory concepts as a basis of the methodology.

2. Narrative, story and the semiotic levels of narrative

WEEV is based on a narrative or story metaphor. However, the exact definition of story and narrative has important implications in the understanding of what "narrative metaphor" means. The current state of the art shows that different definitions or interpretations of narrative and story coexist simultaneously in the literature, but some important aspects can be extracted to provide a useful definition for this thesis.

Video games as narratives

WEEV presupposes that a video game can be treated as a narrative. This isn't, however, the only line of thought currently considered in the literature about video games and narrative. Different scholars provide different definitions of both video games and narrative, questioning the fact of paramount importance for the correct interpretation of the WEEV metaphor: that video games can be considered narrative.

Juul (2005) is one of the main detractors of the concept of "games are narratives". He argues that definitions of narrative have been expanded too broadly as to include anything in the world and thus calling games narrative lacks any significant meaning. However, even if Juul presents 6 different definitions of narrative in which games generally do not fit, he makes specific exceptions in all cases for "progression games" in "fictional settings" (Juul, 2005). Adventure games and simulation *point-and-click* games, the main kind of games created with WEEV (and <e-Adventure> in general), both fit these two criteria so even following the strict definition of narrative provided by Juul we could study the games created with WEEV within a narrative context.

Ryan (2006) presents the conflict between different definitions or perspectives as to the meaning of narrative. Strict definitions only apply when the spoken word is used as medium. However other definitions attribute a metaphorical value to the spoken word, considering any means of transmission of stories as narrative mediums. Some other authors (such as Ryan) consider a definition of narrative based on a series of characteristics, and consider different grades of "narrativity" for different mediums and stories. In this sense, video game can posses high "narrativity" by this could not only be the case.

The position adopted in this work is the second one, where a narrative medium is that which transmits a story. This definition does not allow games to be automatically considered as narratives, and even when they transmit a story, they only became an "instance" of narrative while they are being played.

Besides, we follow a structuralist approach to narrative theory, using a model of several layers of narrative meaning. This model of narrative meanings, or semiotic model, presented in Figure 2, is the one proposed by Lindley (2005). In this model, the "narrated text" has been generalized to any means for the transmission of the narrative.

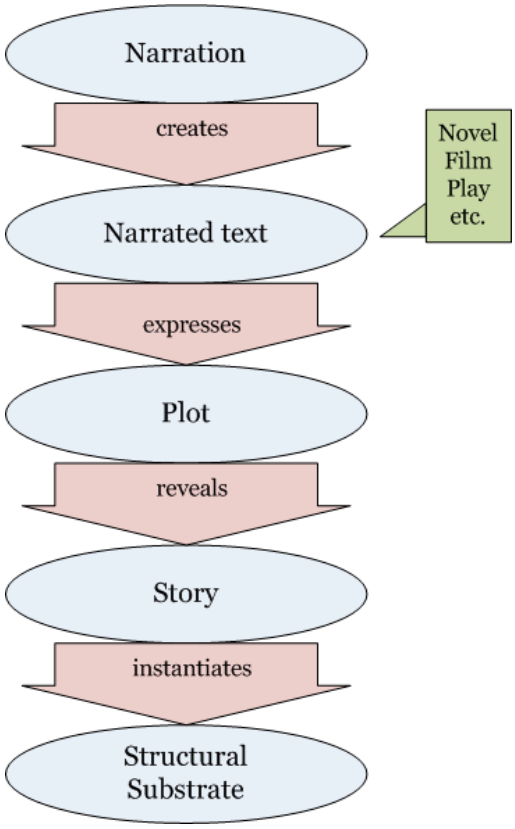


Figure 2 Layers of meaning or semiotic model of narrative texts, according to Lindley (2005)

Video games within the structuralist narrative approach

The use of the phrase "narrative metaphor" arises from the fact that the metaphor is based on narrative theory. However, the narrative itself cannot be represented, as according to the structuralist school of though in narrative theory the narratives are the instances in time that express stories. In the same way, according to Wolff *et al.* (2007) "the story is the collection of facts (such as events, actions, characters, etc.), whereas the narrative relates to the particular way in which these facts are arranged and conveyed to a reader or audience".

Lindley (2005) presents a model of the semiotic levels in computer games and linear narratives shown in Figure 3. The plot in the linear narrative translates to the performance in a game, as "the plot is not something delivered to the player, but something actively created by the player in the interaction with the game system". In the same way, the story in the linear narrative translates to the model in the game, as "the game story is the total implied game world history as determined by the pre-designed potential of the game in interaction with the game play actions of the player".

Semiotics of computer games	Semiotics of narrative
Narration / discourse	Narration / discourse
performance	plot
model / simulation	story
generative substrate	structural substrate
structural substrate	

Figure 3 Relationship between levels of semiotic structures in linear narratives and video games, according to Lindley (2005)

In this work, story will be used to refer to the underlying model of the game: the interactions with the user and reactions of the system. Sometimes, this is referred to as potential stories, given that the model can include different story paths to be experienced as different narratives by the player. Narrative, although sometimes used interchangeably with story, will refer to the instantiation of the story. That is, narrative is the story of the game as perceived by a player.

3. The structural substrate or story structure in video games

Considering video games as narratives implies that video games could inherit the narrative or story structures commonly employed in other media. This entails benefits for the WEEV system, which can provide these structures to represent the stories of educational video games using proven narrative techniques. Current literature shows that games already benefit from these structures, borrowing them from different fields and media.

Lindley (2005) defines the structural substrates (also called story structures) as the general structures underlying the formulation or generation of stories, and from which it is possible to create many different stories. Lindley also finds that "a common example of very specific model of narrative form used in computer games [...] is the three-act restorative structure borrowed from literature, drama and film scriptwriting". The three acts in this structure are:

1. *Beginning*: a conflict is established
2. *Conflict*: playing out of the implications of the conflict
3. *Resolution*: final resolution of the conflict

Lindley argues that when used in games the conflict structure is repeated at different levels of the temporal scale. Lindley also suggests that the structure might not be applied to the game as a whole, using no interactive cut scenes to introduce the conflict (first act) and complete the dramatic arch (third act), resulting in the game play having a limited effect in the story being told.

The most basic of narrative structures, usually referred to just as basic narrative structure, is composed of three parts: introduction, core and conclusion. This simple structure can provide, within educational video games, a framework to introduce concepts (introduction), challenge the player to learn the concepts (core) and evaluate the concepts (conclusion). Besides, this structure could also be used in other ways, depending on the underlying story of the game.

Dickey (2006) proposes that most adventure games (and many other games of all genres) use a *quest* structure. The *quest* is a common narrative structure that appears throughout Western literature and films. Most of the elements of the *quest* might not appear in the same literal way, but are present at least in a metaphorical or emotional equivalent. Vogler (1998) outlined a common structure for the hero's journey in a *quest* in 12 stages:

1. *Ordinary World*: the *hero* is situated in the ordinary world within the story
2. *Call to Adventure*: the *hero* is presented with a situation (e.g. conflict) that requires leaving the *ordinary world*
3. *Refusal of the Call*: the *hero* questions or has reservations about the task at hand
4. *Meeting with the Mentor*: the *hero* meets the *mentor*, who offers advice and guidance
5. *Crossing the First Threshold*: the *hero* commits to the adventure and starts the endeavor
6. *Test, Allies, Enemies*: problems, allies and enemies the *hero* encounters along the way
7. *Approach to the Inmost Cave*: the *inmost cave* is the site of the central challenge to the *hero*
8. *Ordeal*: the central challenge, including a moment where all seems lost
9. *Reward (Seizing the Sword)*: the *hero* survives the challenge and receives the reward of the adventure
10. *The Road Back*: the *hero* starts the journey back the *ordinary world*
11. *Resurrection*: the road back is not without challenges and problems, until the final resurrection of the *hero*
12. *Return with the Elixir*: the *hero*'s journey finishes in the ordinary world with the rewards from the adventure

Besides these stages of the journey, Vogler (1998) also offered guidance on the development of roles. He identified seven character archetypes: *hero*, *mentor*, *threshold guardian*, *herald*, *shape-shifter*, *shadow* and *trickster*. These archetypes can also be found on video games.

The *quest* structure fits well with educational approaches, where the challenges represent the problems, the mentor represents a figure that advices the learner and the main challenge possibly represents an evaluation of all that was learned during the *journey*. Using such a structure is

appropriate for intrinsic learning approaches, where the resolution of challenges in the game have consequences both in the game and the learning outcomes of the student. However, an educational game designer can find it hard to implement, or even be unfamiliar with, such a structure without assistance from the system or expert designer.

All these structures, inspired in classic and contemporary literature and usually found in video games, provide both a theoretical framework to establish a structure in the development process and as examples of the potential use of such structures. Potential educational game developers might not be familiar with such structures, or their use in game development, but the methodology can be tailored to help them develop their stories using structures proven to be helpful in transmitting meaningful and consistent stories.

4. Story telling and narrative theory applied to video games

Story telling and narrative theory has been linked to video games and video game development in different studies. An analysis of the current state of the art allows us to better understand these links and provide elements to improve the design of games with the system and, not least, to consider different uses such as teaching story writing through video game development.

A heuristic was proposed by Dickey (2006) and tested in a real environment. This proposal is based on the *quest* structure. The heuristic consists of the following steps:

- *Present the initial challenge*: the climax in the narrative becomes the problem or project that is the goal for learning in an educational context.
- *Identify potential obstacles and develop puzzles, minor challenges, and resources*: the smaller obstacles and challenges in a story become different procedure, skills and content knowledge that will help learners complete the challenge in a learning environment.
- *Identify and establish roles*: Using the archetypes identified by Vogler (1998) characters and situations must play certain roles in the game. Most important are the roles of the *hero*, usually performed by the learner and the role of *mentor*, which provides guidance to help foster learner reflection, analysis, planning, and evaluation of strategies.
- *Establish the physical, temporal, environmental and emotional, and ethical dimensions of the environment*: These dimensions establish the environment of the game, according to Rollings & Adams (2003). The physical dimension defines the space in with the player moves. The temporal dimension defines the role of time in the game. The environmental dimension defines the appearance of the setting, if it is fantasy or realism, the historical context and geographical location. The emotion dimension describes the emotions of the characters. The ethical dimension defines moral aspects. In an educational context, these dimensions need to be established to support the storyline by reinforcing plausibility.

- *Create a backstory*: Provides an outline for the different dimensions of the environment, as well as a profile of the protagonist. The main challenge or call to action might be introduced.
- *Develop cut scenes to support the development of the narrative story line*: Cut scenes provide ongoing narrative and may be used to deliver key information or plot hooks. In educational contexts, they could provide feedback about the learner's progress and set up the next stage of the narrative.

Research has been conducted by Robertson *et al.* (2004) to study the possibilities of game authoring as a tool to foster narrative development in children. During a workshop experience using the *Neverwinter Nights*™ game and toolset (i.e. a series of tools to create environments and stories for the game) they propose a set of steps to create games using techniques used in other storytelling media. They present a particular instance of these steps in their experience:

- Group discussion about games
- Initial trial of *Neverwinter Nights*™ game
- Character model making
- Plot planning
- Storyboarding with digital cameras
- Game authoring using the *Neverwinter Nights*™ toolset
- Reflecting and planning

These steps as presented by the authors allowed children to develop their own games. The results of this research leads the authors of the study to conclude that the creation of games by students has many potential benefits in the domain of literacy and narrative development. The task of creating video games resulted enjoyable, engaging and rewarding, and well within the reach of 12-15 year olds.

Other comparisons of the relationship between video games and narrative theory or story writing have also been proposed. For instance, Robison (2008) argues that lessons can be extracted from the game development process to improve and re-think composition curricula.

All these studies argue for a strong relationship between narrative theory and video games, and this relationship influences both fields. This *bijective* relationship has two main consequences in this work: narrative theory can be used to improve the development process, helping users build upon proven narrative theory concepts; and the game development process can be used to help students learn story writing concepts.

5. Explicit representation of game story or plot structures

The story or plot definition in WEEV is based in an explicit representation. This explicit representation will fit different structures (tree, linear, etc.) depending on the game the creator is designing. The possible structures that a game can have, however, have been studied in the current literature about video games. A detailed study of this literature provides insights into the descriptive power required by the system.

Ryan (2006) identifies two types of structures or architectures in interactive narratives: those affecting discourse (Figure 4) and those affecting the story (Figure 5). The first structures represent different ways to navigate a fixed story (such as common in hypertext narratives) and the second kind represents patterns of choices that affect the resulting story.

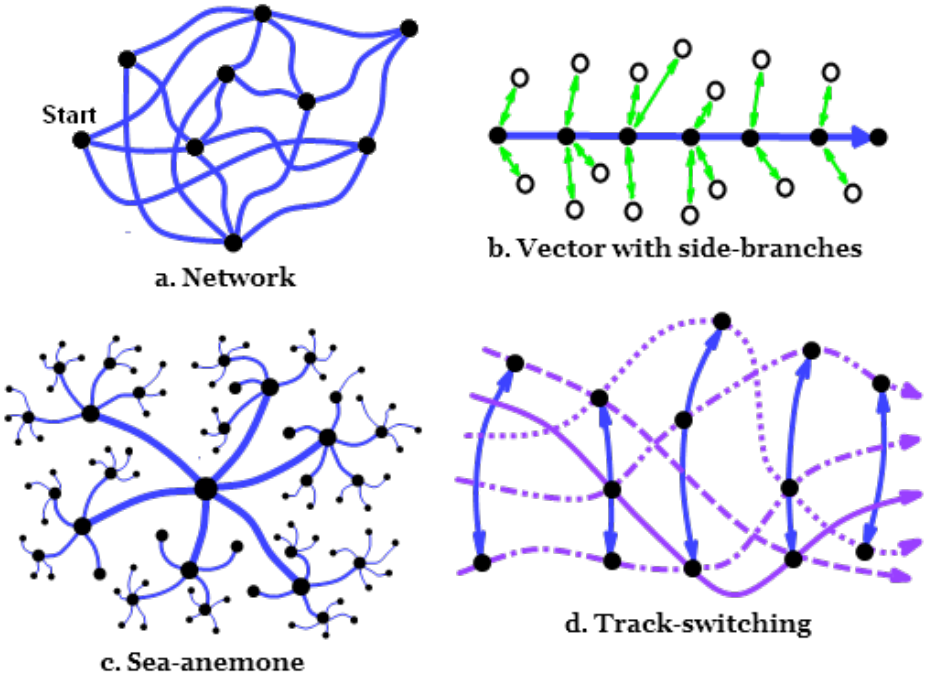


Figure 4 Interactive architectures affecting discourse according to Ryan

The architectures affecting discourse allow a same set of events to be visited in different orders. The underlying story does not change, but the narrative experienced by the user does. The network architecture (Figure 4, a) presents challenges given that some nodes can be revisited though different paths, allowing for incoherent sequences, but is the most common patter for discourse-level interactivity. The vector architecture (Figure 4, b) maintains the sequence as in non-interactive narratives but allows branching events to enrich the story. The "sea-anemone" is a radiating architecture (Figure 4, c) that allows recursive unfolding of the information, widely used to organize web sites; it can also be used in some particular kinds of narratives. Finally, the track-switching architecture (Figure 4, d) allows for variations in the discourse such as those introduce by the network (every node is linked to other nodes) while making it impossible to "travel back in time" thus eliminating incoherent narratives. The various patterns can be combined in the creation of narratives, and can be used in interactive narratives. Although these structures can appear in games, the fact that they do not affect the story

limits the consequences of the player's actions and thus the interest of the game.

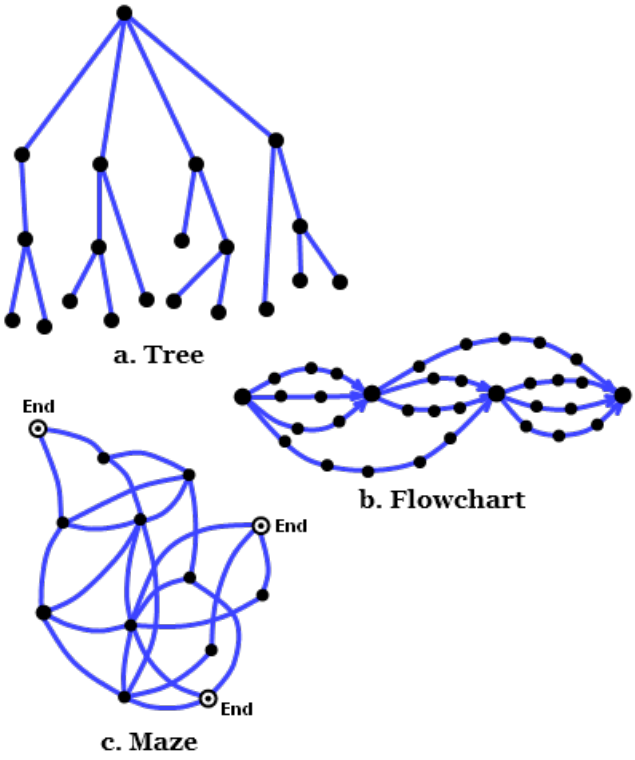


Figure 5 Interactive architectures affecting story according to Ryan

Interactive architectures that affect the story must capture the flow of time. The tree (Figure 5, a) allows different branches to grow at decision nodes but does not allow the story to flow to a previous point. The tree can easily grow unmanageable, however, so it is most efficient for stories with little interactivity. The flowchart (Figure 5, b) limits the proliferation of nodes, making it more manageable, although attention must be placed in the "merging" of strands of story so that they do not allow the introduction of incoherent narratives. This architecture is efficient to organize stories with self-sufficient episodes. The maze (Figure 5, c) is presented as an architecture if it represents a topography of the virtual world, making the itinerary through the labyrinth the story.

From Ryan's analysis of the different architectures of the story we can conclude that the flowchart is the most appropriate for interactive games. Given that the story must change depending on the player's action, the architectures that affect discourse do not apply for this purpose. The other structures affecting story regard the topology, which is usually independent from the story of the game in adventure settings and thus does not apply, and the tree, which is to complex to manage.

Lindley (2005), however, argues that "considering the example of games in which the player's moves map onto low level in-game player character actions, it becomes completely impractical to map out all of the possible story alternatives at a low level". Still, we consider that two aspects of educational video games make this approach practical for this particular case:

1. Educational games are of limited length, attempting to cover only one lesson or particular subject (Pablo Moreno-Ger, Burgos*, et al.*, 2008).
2. Teachers and other educational professionals usually lack deep technical knowledge and can benefit from explicit and visual representations.

Besides, from a technical perspective two other aspects can make practical such a representation:

1. "Low level" can be interpreted in different ways, allowing actions of high semantic value to be defined as "low level" actions in the system.
2. Hierarquical organizations can make complex information easier to understand, encapsulating series of "low-level" actions into "high-level" structures.

These aspects of visual languages are covered by the visual language theory.

6. Domain-Specific Visual Languages (DSVL) and Visual Programming Languages (VPL)

The explicit representation of the story in WEEV depends on a language that can both describe the flow of actions and events (and another for the game world), and be directly and automatically converted into a playable game. Domain-Specific Visual Languages (DSVL) and Visual Programming Languages (VPL) can fit this needs as research shows, and previous experiences have been studied that provide a framework for the DSVLs used in WEEV.

Particularly, DSVL and VPL language theory applies to WEEVL (WEEV Language) the domain-specific visual programming language used to describe the story in games. However, the concepts of DSVL also apply to the language used to describe the virtual world or setting in WEEV game development.

Domain-specific languages arise to provide better solutions to a small set of problems that cannot be achieved using general purpose programming languages. Van Deursen *et al.* (2000) provide a definition: "A *domain-specific language* (DSL) is a programming language or executable specification language that offers, through appropriate notations and abstractions, expressive power focused on, and usually restricted to, a particular problem domain."

Van Deursen *et al.* also point out different advantages and disadvantages of DSL, for example, some of the advantages are:

- Solutions can be expressed in the idiom and at the level of abstraction of the problem domain.
- DSL are concise, self-documenting to a large extent and can be reused for different purposes.
- DSL enhance productivity, reliability, maintainability and portability.

Among the disadvantages, the ones most relevant in regards to the subject matter of this thesis are:

- Costs of designing, implementing and maintaining a DSL, as well as cost of educating the users.
- Difficulty of balancing between domain-specificity and general-purpose programming language constructs.
- Difficulty finding the proper scope for a DSL.

Finally, they also show that all theoretical aspects of the development, implementation and use of DSL have been mostly covered in the literature. The design process involves analysis, implementation and use, and the standard compiler theory covers most aspects of the compilation or interpretation implementation.

DSVL compose a subset of the DSL, with which they share most features. The main differences are the use of visual elements and metaphors and that visual languages are in most cases oriented to end-users with limited technical skills (even though this might not always be the case). As WEEV is presented as a way to develop or program games, the language used can be considered a VPL, or more specifically a domain-specific VLP in contrast to general-purpose VPL (Boshernitsan, *et al.*, 2004). The language used to describe the story, WEEVL, is a domain-specific (educational video games) visual programming language, and as such is defined at the intersection of DSL, DSVL, VL and VPL (Figure 6).

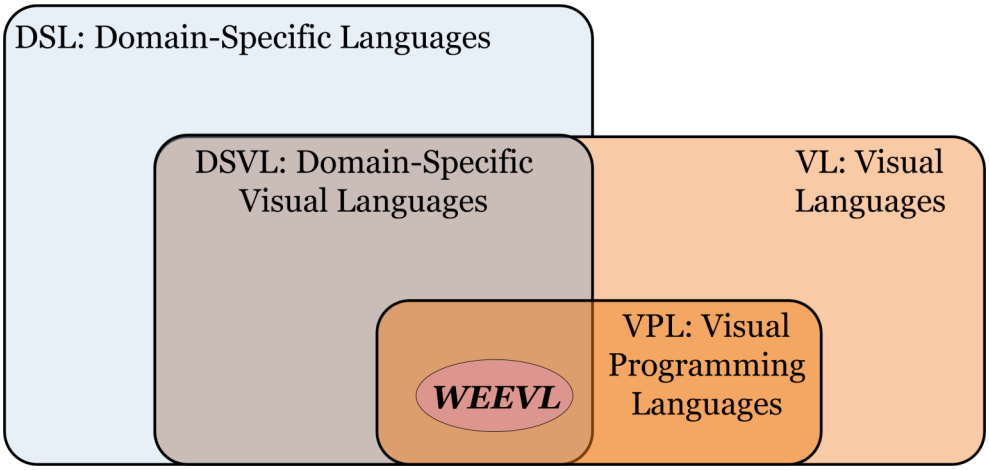


Figure 6 DSL, DSVL, VL, VPL and WEEVL: The intersections between language specification alternatives

Different arguments are presented for the use of VPL. For example, as people think and remember things in terms of pictures reducing the necessity to translate visual ideas into textual representations can reduce the learning curve (Boshernitsan, *et al.*, 2004).

Visual Programming Languages can be subject to different classifications. Boshernistan *et al.* (2004) present the following, not mutually exclusive, categories:

- *Purely visual languages*: rely on visual techniques throughout the programming process. No interim text based language is used.
- *Hybrid text and visual systems*: provide a combination of visual and textual elements. Includes both visual languages that convert to a

high-level textual language and textual languages that rely on graphical elements.

- *Programming-by-example systems*: "learn" how to perform a particular task by the manipulation of visual objects by the user.
- *Constraint-oriented systems*: especially used in simulations, the user models the behavior of objects subject by constraints (e.g. natural laws).
- *From-based systems*: "represent programming as altering a group of interconnected cells over time and often allow the programmer to visualize the execution of a program as a sequence of different cell states which progress through time".

These studies show that this work can benefit from previous research into VL theory to define the language, create its representation and transform such representation into another language (<e-Adventure> games, in this case).

7. Conclusions of this chapter

Educational video games have become accepted as complements to traditional educational approaches. Custom developments, dependant on specific authoring tools that fit the needs of educators, have been proven of interest to achieve better results. The different factors and aspects that influence game design have been thoroughly studied, particularly regarding story and narrative aspects, and are considered of special importance to engage learners and increase the learning outcomes of games.

Studies show that not only video games can be created using a narrative approach, but that a narrative approach to video game development can help in literacy and narrative development of children. This double use of authoring tools can be enhanced by the tighter integration of the narrative metaphor in the game development process.

Different studies in narrative theory have generalized approaches or underlying structures in stories that have the potential to be reused to create new stories. Other studies analyzed existing games, showing that the same structures that are found in novels or films are present in successful commercial games. Besides, the same approaches used in narrative studies have been directly applied to games, identifying the underlying architectures, stories or plots that define the instantiation of the narrative by the player.

The generalized structures underlying the plots or stories of games are usually represented using visual constructs, and such is the field studied by Domain Specific Visual Languages (DSVL) and Visual Programming Languages (VPL). Different proposals of this field allow the direct implementation of the story of games, using approaches that have been validated for other uses.

VPL theory provides a framework for the definition of a language to describe stories in games, including different mechanisms to process, interpret or compile such a language to do useful work. At the same time, this approach has been validated to suit the needs of users with limited technical knowledge and to provide a learning curve that is less steep than that of textual programming languages.

CHAPTER III. RELATED WORK AND APPLICATIONS

This chapter studies different video game authoring platforms, which were selected for their relevance in the educational field or the size of their user base. Other authoring tools, meant for contents other than video games, were included when their approach was considered interesting in relation to this work. The <e-Adventure> platform, which provides a framework for the system introduced in this thesis, is described in detail.

1. Video Game development platforms

Nowadays, several video games development platforms are available for end-users and professionals. Some even have *light* or free versions and paid alternative for professionals and corporative users. Few of these tools, however, incorporate educational features out-of-the-box. The games that can be created with each platform vary, as does the platforms and devices where the resulting games can be played.

The following tools where selected as representative sample of those available and does not intend to be exhaustive. The platforms studied are: *Adventure Game Studio* and *Adventure Maker* as representatives of adventure or *point-and-click* game-authoring tools, *Unity*™ as a representative of professional and semi-professional tools with freely available versions and *Thinking Worlds*™, as it includes a particular approach to game creation and is focused on serious or educational video games.

Adventure Game Studio (AGS)

Adventure Game Studio (AGS) is a widely used game-authoring tool, developed to create *point-and-click* adventure games. This tool is freely available at the official website[4], where free games developed using it are also regularly uploaded. It uses a content-based approach to game creation, based on the defining of the different components of the game (content) and the interactions between them, implemented in a graphic user interface (Figure 7). *AGS* includes a scripting language to describe complex interactions, as well as a debugger, support for different media (OGG, MP3, etc.), language translation support and a plug-in system.

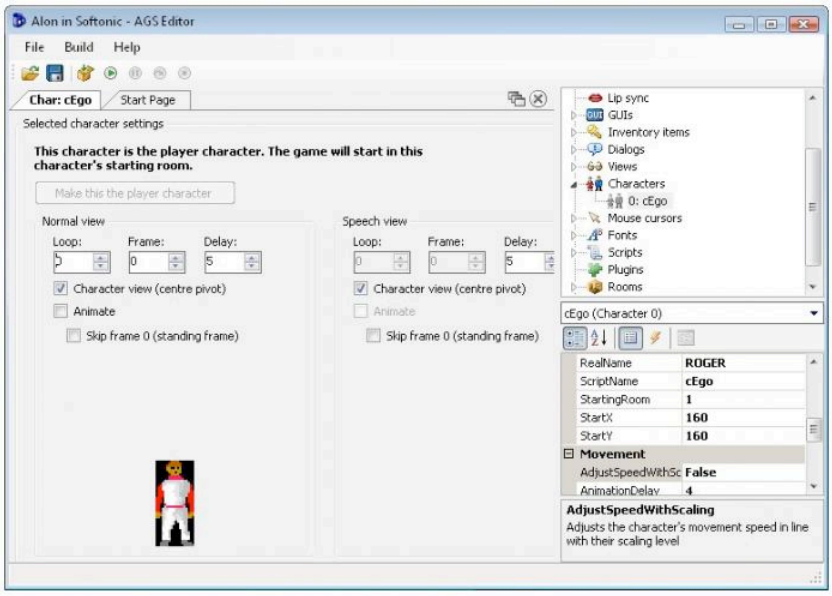


Figure 7 Screen-shot of the *Adventure Game Studio* authoring tool

[4] *http://www.adventuregamestudio.co.uk/*

Adventure Maker

Adventure Maker is another widely used game-authoring tool, freely available on the official website[5]. Support for some advanced features such as plug-ins require a pay-version of the system. Games developed for using this tool are freely available on its website. The tool uses a general approach to "adventure" game creation, proposing different types such as virtual tours, guided visits and interactive presentations. The system uses a graphic user interface (Figure 8) and a scripting language to describe the most complex interactions. Current versions allow games to be created for different target platforms besides PCs, including *Apple*™'s *iPhone* ™ and *Sony*™'s *PSP* ™. This tool does not included specific educational features (standard compliance or assessment mechanisms) although the developers suggest that it could be used for some educational purposes.

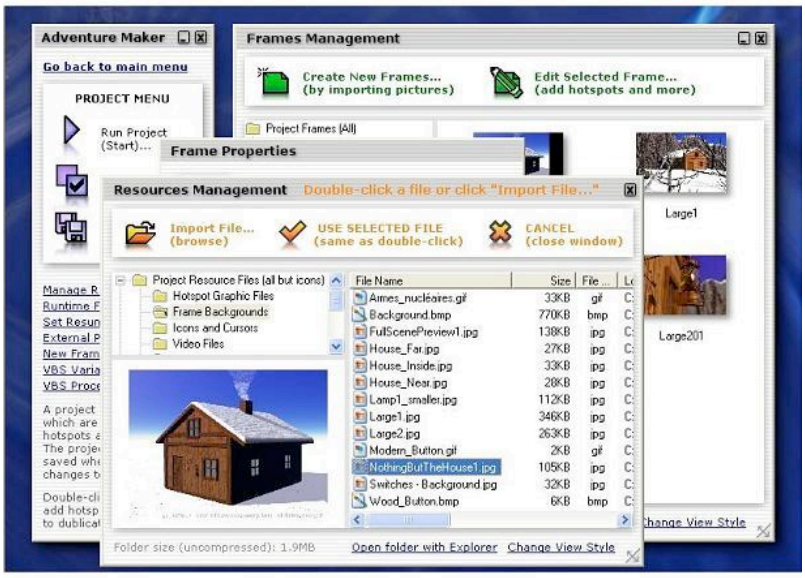


Figure 8 Screen-shot of the *Adventure Maker* authoring tool

Unity

Unity™ is a professional-grade video game development tool. A free version (with some limitations) is available at its website[6]. This tool was originally designed to create 3D PC games, but has evolved to include *Nintendo*™'s *Wii*™, *Apple*™'s *iPhone*™ and other mobile devices as target platforms.

Unity™ provides a complex but extremely powerful authoring environment, as well as a highly optimized engine for all the supported platforms. This tool is widely used in professional and semi-professional games publicly available.

[5] *http://www.adventuremaker.com/*

[6] *http://unity3d.com/*

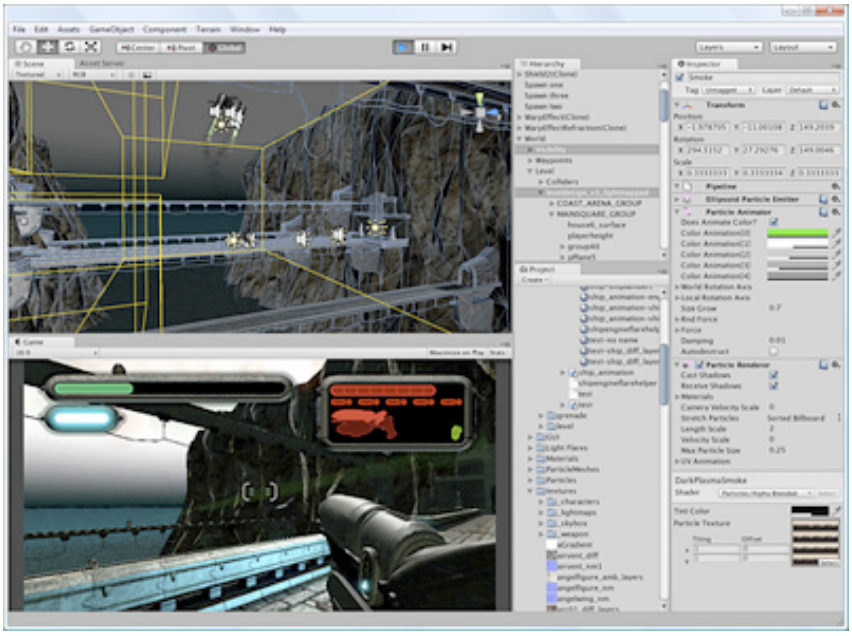


Figure 9 The 3D authoring environment of *Unity*™ provides powerful features using a complex user interface.

Thinking Worlds

Thinking Worlds™ is an authoring tool specifically designed for education. A demo version is available at its website[7] although different production versions (training courses and educational facilities) as well as the 3D art library require different licensing fees. This tool allows the creation of 3D interactive worlds, using a 3D editor that allows the combination of existing resources to create rich worlds (Figure 10).

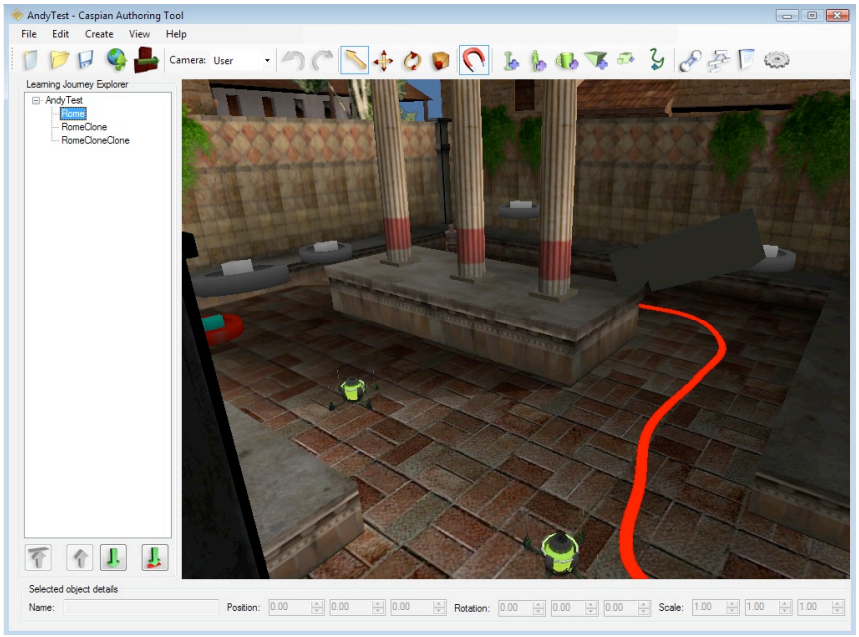


Figure 10 The *Thinking Worlds* 3D environment authoring tool.

[7] *http://www.thinkingworlds.com/*

Although the interactivity of the resulting games is rather limited, the flow editor presents a new approach to game development (Figure 11). This flow editor uses a *hybrid text and visual system*, where the flow is represented graphically but the different elements within this flow use a textual representation. This approach results in a rather complex representation, but at the same time allows many specific details of the different elements to be edited directly.

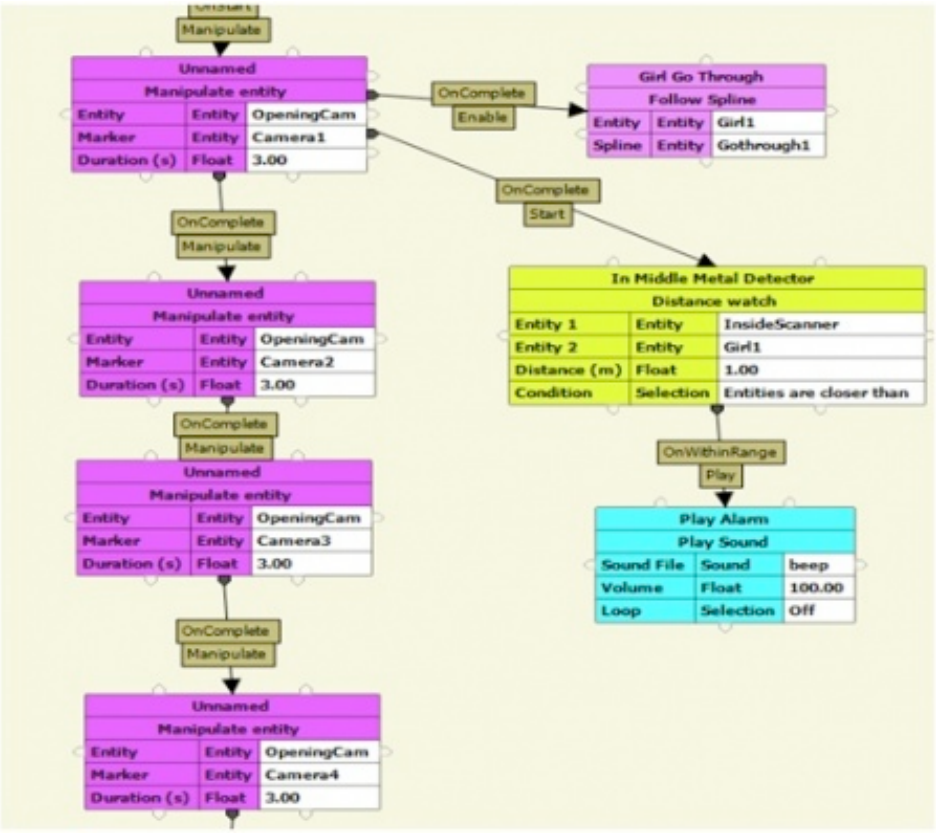


Figure 11 The "Storyboard" editor in *Thinking Worls*™ allows the definition of the flow of the game, mostly concatenating predefined actions and resorces modifications (camera positions, NPCs, etc.)

StoryTec

The *StoryTec* platform is introduced as a "Digital Storytelling platform for the authoring and experiencing of interactive, non-linear stories" (Göbel*, et al.*, 2008a, 2008b). This platform is not currently available, but introduces storytelling concepts into the game development model. The *StoryTec* platform includes both an authoring environment and a runtime engine.

The authoring platform, based on a pluggable framework, is composed of different parts including a "Story Editor" that is used to manage the story structure, a "Stage Editor" to edit scenes of the game, an "Action Set Editor" that is a visual editor of high-level story logic in a per scene basis and an "Asset Manager" used to import different assets into the games.

2. Other tools based on narrative or story writing concepts

Other tools, developed for general or specific purposes other than games, use narrative or other similar metaphors to describe games. Besides, other tools use DSVL to describe stories, interactions or other elements in the systems. Here we present some of the most interesting tools that fit these criteria.

Storyspace

Storyspace was a system, developed in the late 80s, used to create interactive hypertext stories. *Storyspace* used a graphic user interface (Figure 12) based on a state-transition diagram where the main "text body" was placed in the nodes and the transitions represented the actions available to the player at each point in the story. This tool did not produce games, but hypertext narratives even if some of them followed a game metaphor where there was a "good" ending to be found. In most cases, this tool was used to create stories with different branches or different character perspective, allowing the user to choose how the story was to be read.

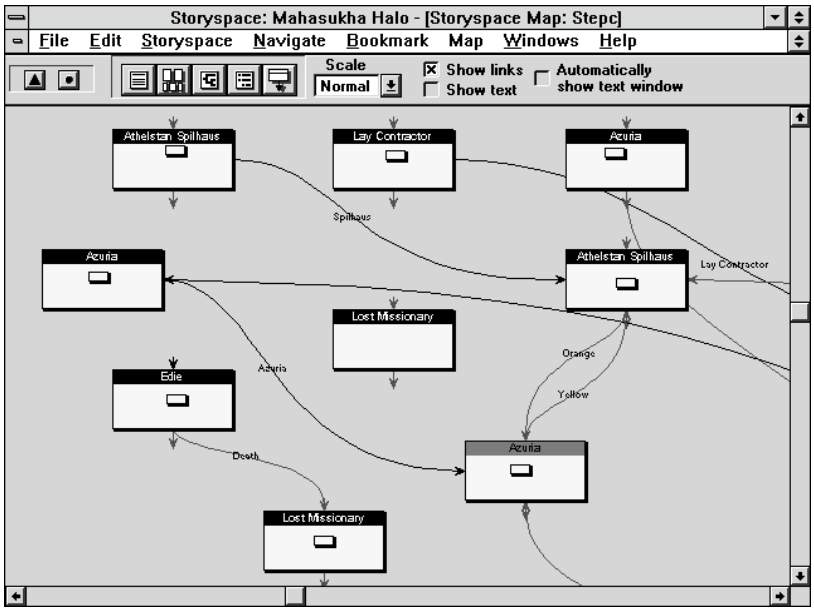


Figure 12 Screen capture of the *Storyspace* story map edting tool

Storyspace, with its graphic representation, eased the creation of hypertext narratives for users with no programming knowledge. Some hypertext stories of considerable success where created with this tool. It was mostly abandoned in the mid 90s with the rise of the World Wide Web and HTML.

Adobe Flash

Probably one of the mostly widely used content creations tools in the Internet, and widely used for games and other interactive and narrative software, *Adobe Flash*™ uses a simple "movie" metaphor to represent the contents. This metaphor is implemented by the use of a time-line that represents the flow of the animation (Figure 13, a). An advanced scripting language (*Action Script*™) and other general-purpose software development tools (e.g. buttons, text inputs, etc.) complement this metaphor. Trial

versions of the software are available, but public or professional developers must buy full licenses[8].

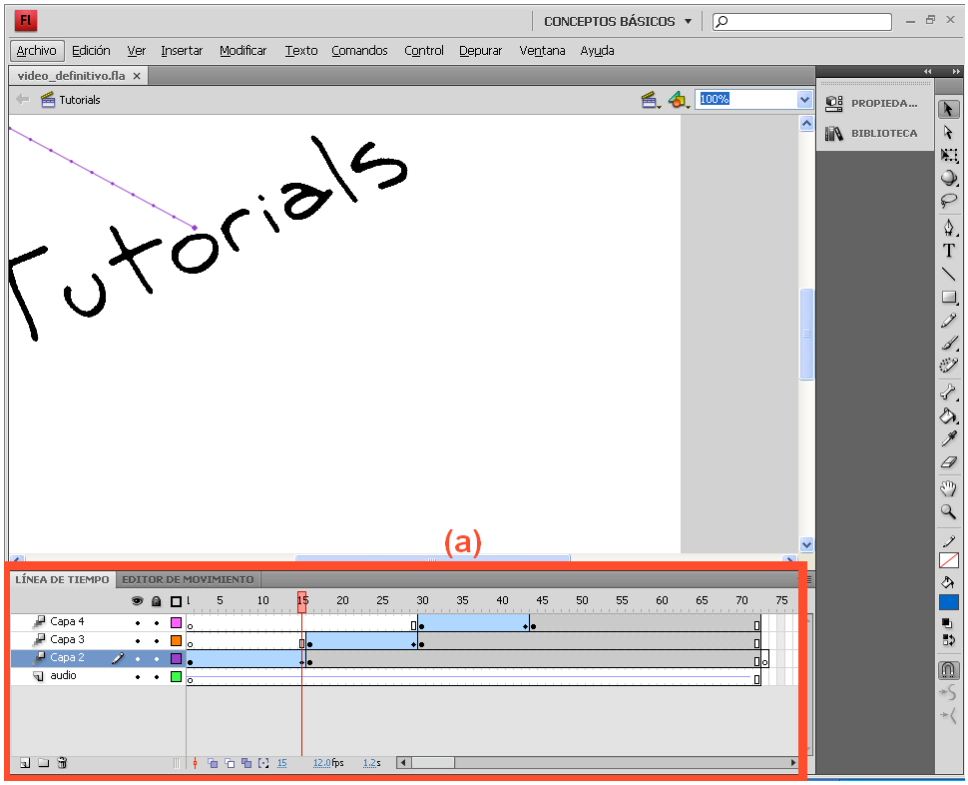


Figure 13 The *Adobe Flash* ™ editor. The time-line (a) is highlighted.

Adobe Flash™ is used to create video games in general and serious games in particular, and is wildly used in custom developments. The wide reach of *Flash*, the broad support (it is one of the *de facto* standards of the Internet) and the easy to use and well know tools make it ideal for medium cost solutions to graphic software developments such as games.

Ren'Py Visual Novel Engine

Ren'Py Visual Novel Engine is a free authoring tool and engine for "visual novel" creation. This tool is available at the official website[9] and games developed with it are cross-platform and can be free or commercially distributed. The website includes a list of different games created using the system as well as tutorials and other helpful information for visual novel development.

Visual novels (basically hypertext narratives with graphic content) are usually created using manga-style drawings, although any 2D graphics can be used. The stories in these novels are developed using a simple scripting language, which allows the creation of the dialogs and the modification of the graphics shown to the user. *Ren'Py* uses a scripting language based on python and includes a full text editor with helper functions to create the scripts of the games (Figure 14).

---

[8] *http://www.adobe.com/products/flash/*

[9] *http://www.renpy.org/wiki/renpy/Home_Page*

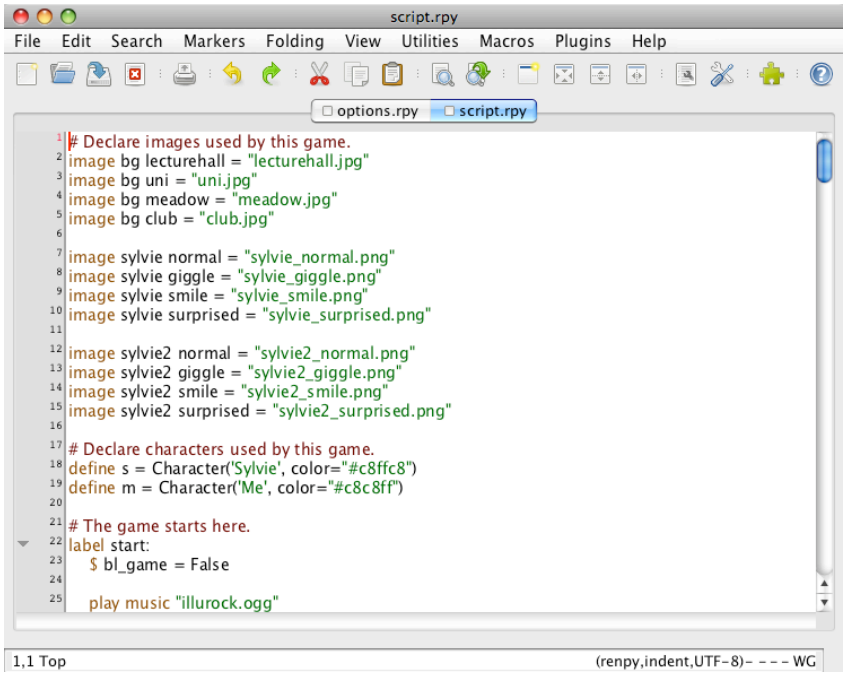


Figure 14 *Ren'Py* authoring platform visual novel script editor

Although these visual novels aren't usually considered games as they rely heavily on text, with limited interaction from the user, the *Ren'Py* platform allows the creation of simple simulation games using the scripting language. However, the interactions are limited to selections of choices in multiple-choice style questions (Figure 15).

Figure 15 *Ren'Py* visual novels use multiple-choice questions to allow user interaction with the story

Storytelling Alice

Storytelling Alice[10] is introduced as a modification of the *Alice 2*[11] software. *Alice* is a platform designed to teach programming using drag-and-drop and simple objects (Figure 16). *Storytelling Alice* modifies *Alice* making it more suitable to create animated stories, as a way to introduce middle school girls (more interested in animated movies than programming) into programming and computer science in later life (Kelleher, *et al.*, 2007). A usable version of *Storytelling Alice* was developed for *Windows* ™ and is freely available online. The language used to describe the stories in both versions of Alice can be defined as a *Hybid Text and Visual language*, as visual components are drag to crate the program, which can also be read as a text.

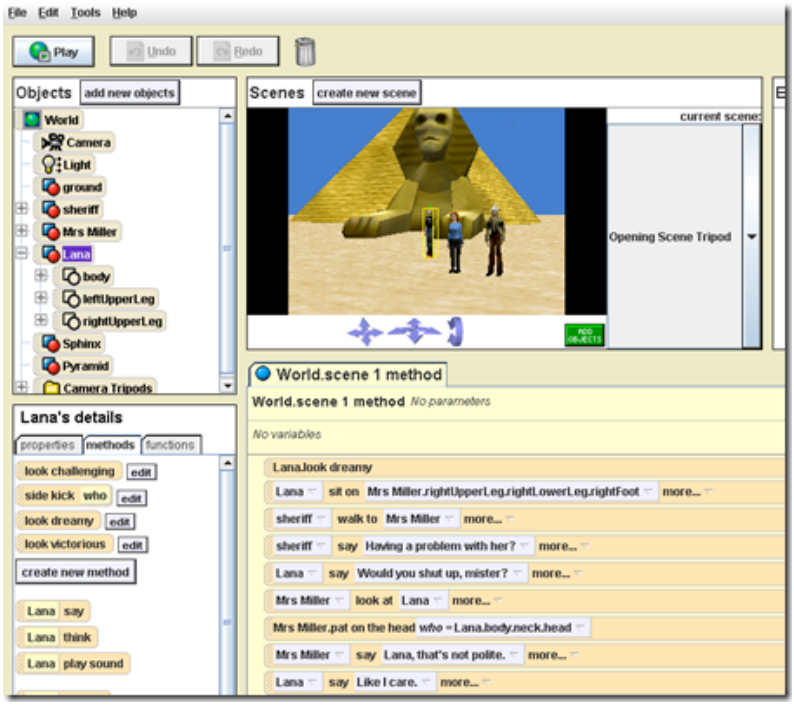


Figure 16 *Storytelling Alice*'s interface allows for the creation of animated movies using drag-and-drop elements while teaching programming concepts

Storytelling Alice is not destined to create video games but animated stories, however, the concepts introduced allows the expression of certain elements in a way closer to storytelling and thus makes it more attractive to users with no programming knowledge. Besides, *Storytelling Alice* introduces a story-based tutorial named "Stencils" that introduces concepts gradually to users while creating a story with the system. Ideas introduced in *Storytelling Alice* were incorporated into *Alice 3*, for which a beta version is available on-line.

[10] http://www.alice.org/kelleher/storytelling/index.html

[11] http://www.alice.org/

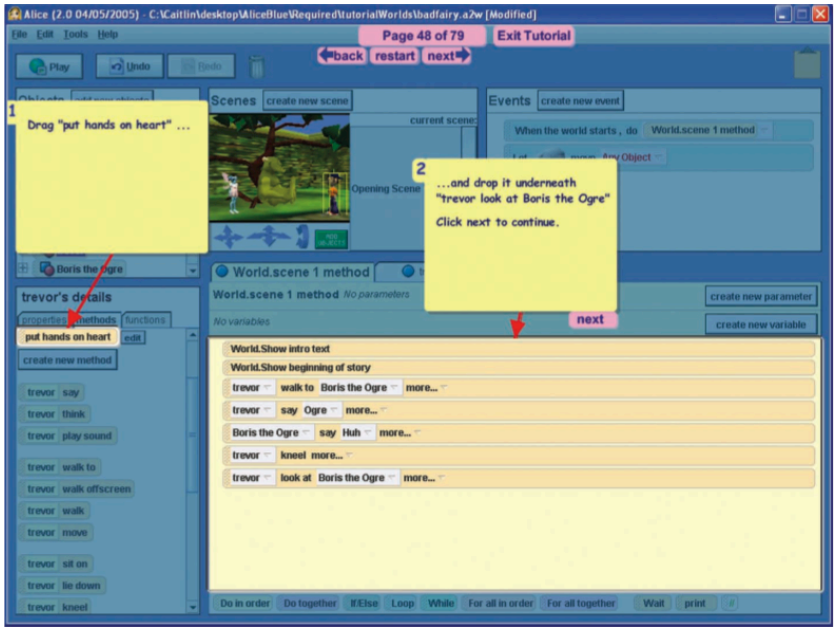


Figure 17 The "Stencils" story-based tutorial in *Storytelling Alice* introduces students to concepts one by one

Crickets and Scratch

Crickets and *Scratch*[12] are two systems developed to support what Resnick (2008) calls the "creative thinking spiral" (imagine, create, play, share, reflect, imagine, and the spiral repeats). These systems present a different approach to content (*Scratch*) and machine (*Crickets*) creation, tailored for children and using different visual metaphors to define the behaviour of elements.

Scratch is created to allow children to create interactive stories, games and animations to be placed in the Web. *Scratch* uses a visual metaphor based on the connection of blocks, which allows the creation of programs without the need to learn punctuation or syntax (Figure 18). *Scratch* is freely available on-line and includes a social component by allowing children to share their creation and build upon the creations of others.

Figure 18 A simple program logic created using *Scratch*

[12] *http://scratch.mit.edu/*

Raptivity

Raptivity™ is a commercial system for interactive content creation[13]. This system uses a unique approach to content creation that is fast and simple, while sacrificing some flexibility. The creation of content in *Raptivity*™ is based on "Interactivities" or predefined programs that can be customized by the user. The customization of "Interactivities" takes place within a wizard-like environment.

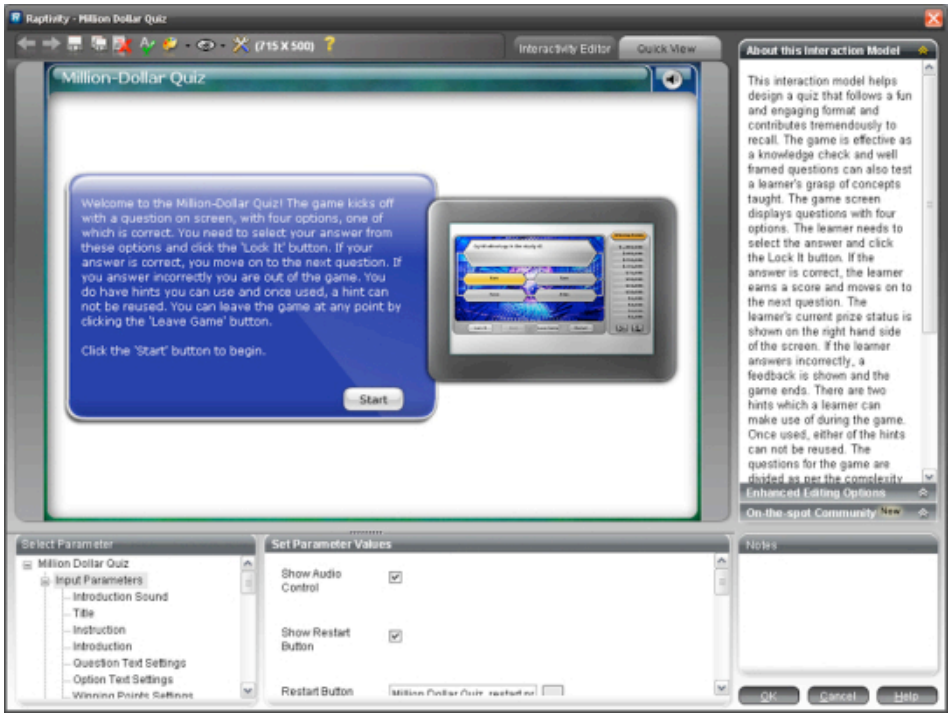


Figure 19 Edition of an "Interactivity" in the *Raptivity*™ system

Raptivity™ includes hundreds of different "Interactivities", including puzzles and other simple games. The results of a customization can be exported as SCORM compatible *Flash* objects.

[13] *http://www.raptivity.com/*

3. The <e-Adventure> Educational Video Game Development Platform

The language we present in this paper was created in the context of the <e-Adventure> platform[14], an environment that aims to facilitate the widespread adoption of educational games by lowering barriers such as high development costs, programming complexity and integration challenges (Pablo Moreno-Ger, 2007). The platform includes a set of game authoring tools, a runtime-engine that can run in standalone mode or integrated in a website and a set of exportation profiles that allow games created with the <e-Adventure> platform to be packaged for different e-learning platforms such as *Moodle*™, *Blackboard*™ or *Sakai*™ (Del Blanco, *et al.*, 2009).

The original <e-Adventure> development model is based on a Domain Specific Language (DSL) and requires the creation of XML files to describe a video game (Pablo Moreno-Ger, *et al.*, 2007). These XML files describe all elements that take part in the game (e.g. objects, scenes, actors, etc.), their associated resources (e.g. images and videos) and their interrelations. The current version of the <e-Adventure> platform includes a visual editor to simplify the game authoring process (Figure 20).

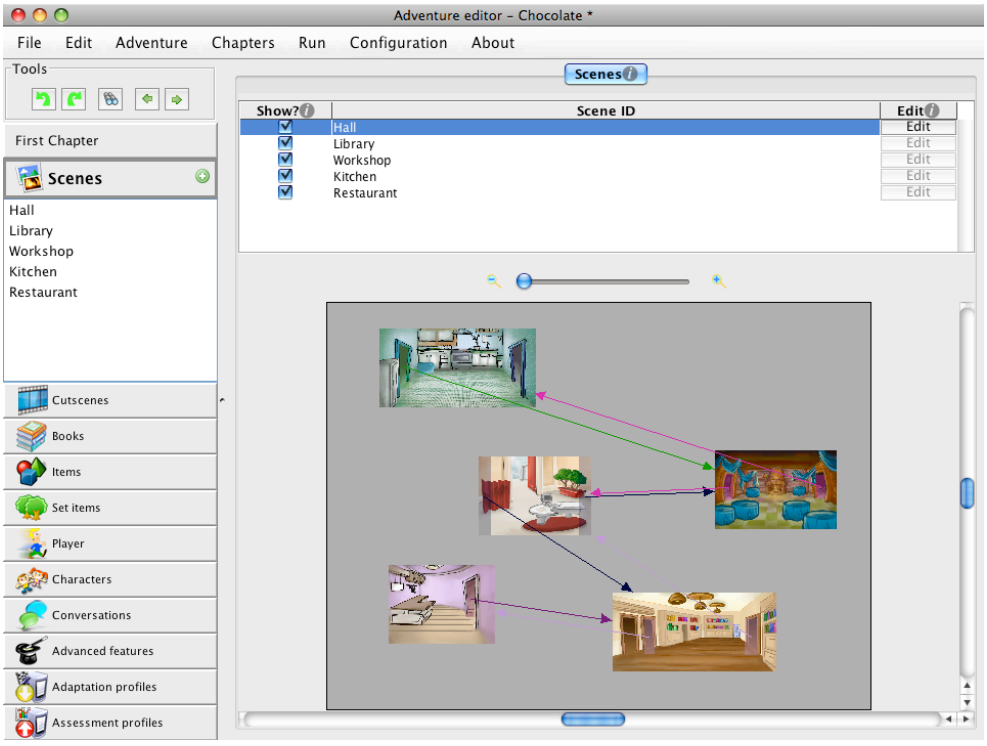


Figure 20 Screen-shot of the <e-Adventure> educational video game editor

The behaviors in <e-Adventure> are defined by the possible actions the player can perform with an element (e.g. transit through a door, grab an item) and the conditions that establish the availability of actions or interactive elements in the game world. The consequences of those actions are called effects. The effects in <e-Adventure> can be either perceptible (e.g.

[14] *http://e-adventure.e-ucm.es/*

launch a conversation or move a character in the scene) or transparent for the player (make changes in the aforementioned conditions and the internal state of the game).

Game states are implicit and internally represented as a set of values for variables and flags. <e-Adventure> games can be represented as implicit Finite State Machines (FSM): on the one hand, the values of variables and flags at a given point of the game are the states; on the other hand the actions of the player in the game can be represented as the state transitions as they trigger changes in these variables and flags.

A downside of this content-centric approach is that the story *emerges* from the different behaviors and is not explicit in any part of the development process. Besides, this implicit story-flow is hard to understand and makes it difficult to develop complex storylines while involving different experts in the process. Our intention is to invert the authoring process and make the implicit FSM visually explicit, by introducing an appropriate visual metaphor that allows authors to edit and understand their story directly.

4. Conclusions of this chapter

As shown by the systems studied in this chapter, different approaches to game creation in general (and educational game creation in particular) have been proposed and implemented. No solution has proven to be a silver bullet that can address all problems that users can find, and it cannot be expected that the approach presented in this work be such a solution. However, knowing the advantages and disadvantages of other systems has proven useful to identify both the things where WEEV can be of most use and different accepted solutions to particular problems.

The approach of incorporating a narrative metaphor in the content development process has been successfully implemented in other systems. However, the fact that these systems are created for other uses (either more general or more specific) leaves the field of educational video games open to the implementation of such an approach. Besides, we believe that most solutions do not take the metaphor far enough to take full advantage of its potential in reusing previous story writing knowledge to develop educational games or to be used to teach story writing concepts to students by its use.

CHAPTER IV. WEEV: WRITING ENVIRONMENT FOR EDUCATIONAL VIDEO GAMES

This chapter describes the main aspects of WEEV. WEEV is introduced both as a methodology for game development and an implementation of this methodology. A description of the wizard and other elements is provided.

1. WEEV: Writing Environment for Educational Video games

WEEV (Writing Environment for Educational Video games) is both a working game development environment and its underlying methodology. The aims of WEEV are three-fold:

- Make the development of educational video games as easy as possible
- Create a description of educational games that is useful for domain-experts to understand, evaluate and validate
- Help the developers place more importance in the underlying story of the game

To achieve these aims, an approach inspired by the heuristic proposed by Dickey (2006) has been generalized into an educational video game methodology that is the central piece of the WEEV system. This allows a structured approach for the development of the game to be implemented sequentially and provide feedback to the user as a guide to achieve better results.

Different parts of the games will be described either using plain language or through Domain-Specific Visual Languages (DSVL). These approaches reduce the complexity of the information presented to the user, as the DSVL provide a graphic representation that is easy to understand. The game or virtual world uses such a representation to describe the "places" available to the player. The story uses another DSVL named WEEVL (WEEV Language) that will be described in detail in the next chapter.

The WEEV methodology

There is no widely accepted standard set of steps that can be used to create a story, but some can be inferred from different studies of the narrative process. Besides, different experiments with students help understand steps that can assist in this process even if they are not always required. From these proposals we extract a methodology that can be applied to the creation of educational video games.

Following the identification of the elements of interactive design by Dickey (2005), an explicit representation of each is included (Figure 21). The *setting* is identified as the world within the WEEV methodology, where the users will define the virtual environment for the game. The *roles and characters* are identified as actors, explicitly establishing the interactive objects and characters of the game. A*ctants* was proposed as a more appropriate definition according to narrative theory, but informal evaluation proved that most potential users were unfamiliar with the word and the concept, so the word *actors* was used in its place. Finally, *actions, feedback and affordances* in Dickey's theoretical framework are identified as the story of the game in the WEEV methodology.

Elements of interactive design (Dickey 2005)	WEEV elements
setting	world
roles and characters	actors
actions, feedback and affordances	story

Figure 21 Elements of interactive design Dickey (2005) and their correlation with elements within the WEEV methodology

This methodology and its elements can also be related to the heuristic proposed by Dickey (2006) (Figure 22). Still, the approach is an approximation (as can be inferred from the figure) and not a direct implementation. However, the framework provided by the heuristic helps encourage a complete and coherent development of games and their underlying stories.

Heuristic approach (Dickey 2006)	WEEV methodology
Present the initial challenge Identify potential obstacles and develop puzzles, minor challenges and resources	Tasks left to the creator
Identify and establish roles	Define the actors in the game
Establish the physical, temporal, environmental and emotional, and ethical dimensions of the environment	Define the world
Create a backstory	Create the story
Develop cut scenes to suppor the development of the narrative storyline	Define the graphic resources

Figure 22 Approximate correlation between the heuristic proposed by Dickey (2006) and the WEEV methodology

Moreover, this approach requires of an explicit representation of the story that is achieved by the use of a DSVL that is able to represent stories as proposed by Ryan (2006) or Lindley (2005). This representation, in contrast with others presented as theoretical frameworks, must be automatically converted to a playable video game.

The WEEV development model

The WEEV development model complements and extends the <e-Adventure> model (Figure 23). In the WEEV system, the user is presented with a series of guided steps in the form of a wizard. After the initial wizard, the user can define the actors, the world and the story of the game. At any time, the WEEV game can be converted into a <e-Adventure> game and edited in the traditional <e-Adventure> editor. Any changes to graphic resources (including graphic assets and positions in scenes) will be stored back in the WEEV game, so posterior conversions will still maintain the changes. Finally, games can be play-tested in the <e-Adventure> engine both directly from WEEV (if the graphic assets are correctly defined) or after

converting them to <e-Adventure> games are editing the finishing details in <e-Adventure>.

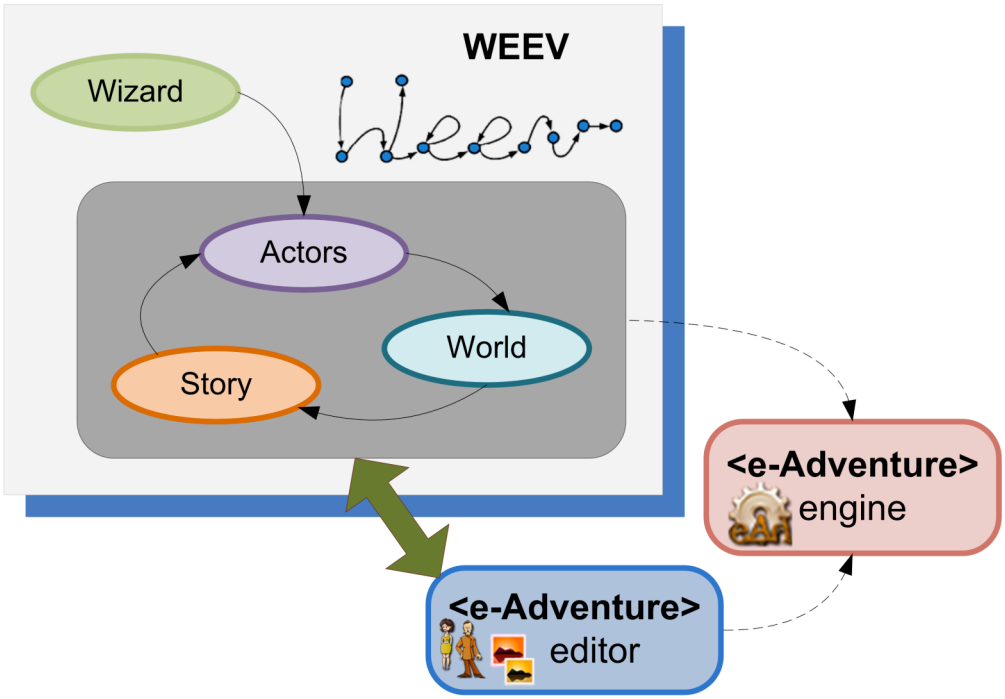


Figure 23 The WEEV development model with the context of the <e-Adventure> platform

This development model allows users already familiar with the <e-Adventure> editor to use it to define elements that are not defined in a straightforward way within the WEEV metaphor (i.e. graphic assets). Besides, it allows the full potential of both tools to be used in conjunction, supporting a more efficient approach.

2. The Wizard

A wizard is a system that takes the user through a number of steps needed to perform a certain task. Wizards are usually the preferred "user-friendly" method for the creation of new content (e.g. adding a new graph to a spreadsheet). The "step-by-step" approach it imposes is useful because it can explain each of the decisions that the user has to make and it divides a complex problem in easy to understand sub-problems. Besides, wizards have been identified as an architectural design pattern that improves the *learnability* of a system (Folmer, *et al.*, 2004), where *learnability* is defined as how quickly and easily users can begin to do productive work with a system and the ease with which they remember how to use it (Ferre, *et al.*, 2001).

The wizard is made up of a number of steps users must go through to create an educational game. The steps are based on the WEEV methodology, although some of the steps of the wizard will just deal with the strictly "game-related" details such as the kind of game or its adaptation capabilities. The steps in the WEEV wizard are:

1. *Starting the game*. The user is asked to give the game a name (i.e. title) and choose the work-folder in the system (Figure 24, a).

2. *Game type*. The user is presented with the different kinds of games available in the platform and asked to make a choice between "Adventure game" (i.e. a third person *point-and-click* adventure game) and "Interactive photo-realistic world" (i.e. a first person *point-and-click* game-like simulation). This choice is made early in the process as the type of the game is important to define how the story develops (Figure 24, b).
3. *Story structure*. Narrative structures can be powerful organization aids in the development of a story, however different games might need different story structures or even none at all. The user is presented with some choices as well as the possibility of defining a custom structure. The default choice, for instance, is the basic narrative structure (i.e. introduction, core and conclusion) (Figure 24, c) or the user could choose a three-act restorative structure (i.e. beginning, conflict and resolution).
4. *Adaptation*. By default, the games will be the same between different game runs and for different users. However, the WEEV system allows game authors to choose one of the predefined adaptation structures that will change the game depending on the users needs (i.e. easy, normal and hard difficulty levels) or the number of times it has been played (Figure 24, d).

Figure 24 Different steps in the WEEV wizard: (a) welcome step; (b) game type; (c) story structure; and (d) adaptation structure

5. *Actors*. The author is asked to add new actors one by one and prompted to describe the characteristics and personalities of these actors. The definition of the actors helps to develop a more consistent and engaging story (Figure 25). Three kinds of actors are available: Items, NPCs (Non-Player Characters) and Parts of scenes. Each NPC or Item actor has a list of appearances, which allow the game designer

to change the visual aspect of the actor mid game by replacing one appearance (and its set of graphic resources) for another.

Figure 25 The actor edition step in the WEEV system. It allows the creation of the different kinds of actors, edition of existing ones and the definition of details for each

6. *World*. The author is given a chance to define the world where the game takes place. This world is defined, in terms of spaces and links among them, as the navigational environment where the action occurs. This step is further detailed in section 3.
7. *Story*. In this step the user will develop the story with the help of the WEEVL visual language. As a consequence of the design of the system, once this step is reach the user will only be allowed to edit the elements of steps 5, 6 and 7 but will not be able to go back in the wizard. This step is presented later in further detail.

3. Creation of the World

The world where the game takes place is an important aspect of the story, as it defines the physical dimension of the game (Rollings, *et al.*, 2003). An appropriate world definition before the creation of the story is helpful to achieve better results (Dickey, 2006) as it can be used as a framework for the events that take place. This process allows the creator of the story to describe the interactions of the character having in mind the context where they will be taking place, thus achieving a deeper integration of the world and the story.

In WEEV the world is described, by the use of a DSVL, as a set of interconnected spaces or scenes through which the player will move (Figure 26). Each space has a name and so does every connection from one space to the other. The user has also the possibility to place the actors in the spaces where they will appear in the game using the same representation.

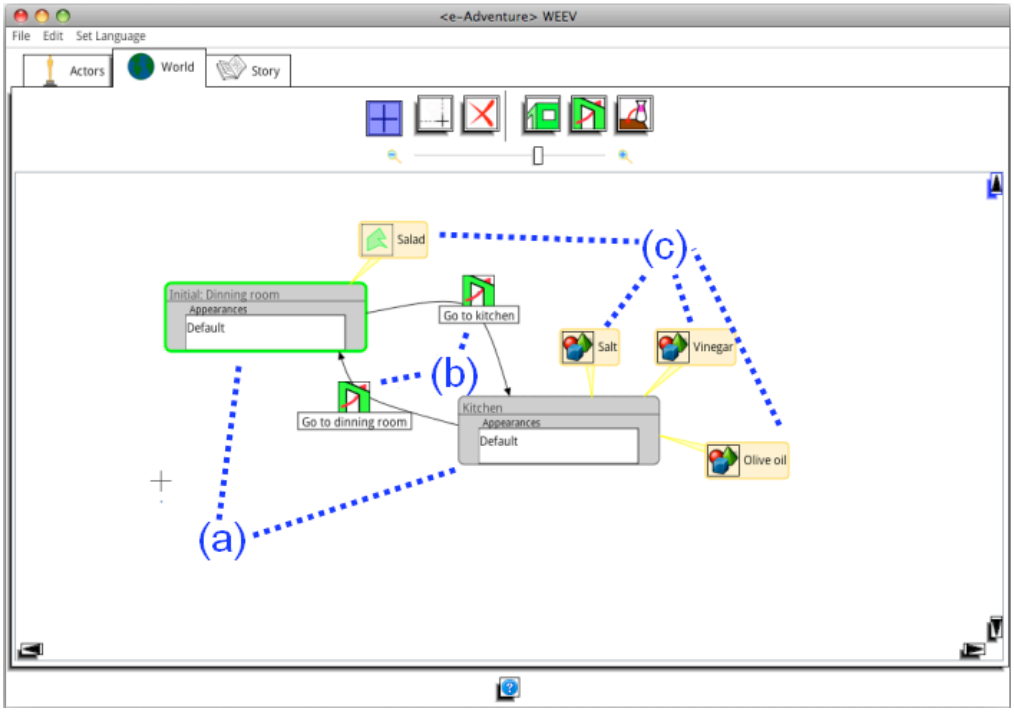


Figure 26 The world edition panel uses a DSVL with different elements: (a) world spaces; (b) space links (exits from one space to another); and (c) actor placements (indicate in which space each actor will appear)

The world, defined using this DSVL can be automatically converted to an equivalent <e-Adventure> representation, based in "scenes", "exits" and other necessary elements of the target language. Besides, in this same panel the graphic details (e.g. the exact position of elements) can be edited using a dialog accessible using a contextual menu.

The spaces in the world have a list of appearances. These appearances allow for the designer to change how the player views the space during the game. For example, a "Public square" space might have "Day-time" and "Night-time" appearances, each with different graphic resources, that can be swapped during the game.

4. Creation of the Story

The story is a central part of any game, but this is especially true in adventure games and other *point-and-click* games (Kiili, 2005); and educational games (Dickey, 2006; Malone, 1981). However, existing tools require a high degree of technical knowledge to understand the story. This is a consequence of the story being implicit and described as the relation between different game elements through the use of complex Boolean logic conditions. This approach is usually referred to as content or object-oriented game development and is used by most currently available frameworks (e.g. <e-Adventure>, *Adventure Maker*, *Adventure Game Studio*, etc.).

In story writing, however, the story is made explicit by the words that describe it. This same approach of explicitly describing the story is the one adopted in the WEEV system. Some programs do this by the use of a scripting language (e.g. *Ren'Py Visual Novel Engine*) but in WEEV a DSVL is used to define a story as a state machine. This approach is easier to

understand and improves the *learnability* of the system. The changes in the story will no longer be determined by Boolean logic but will be explicitly represented as part of the language as transitions between states.

WEEVL, the language used to describe the story, is detailed in the following chapter.

5. Educational features

Educational features in the WEEV system are introduced in games using the same metaphor as the rest of the system. In contrast, other platforms require educational features to be defined in a separate step, using a different metaphor or outside the context where they are meaningful to the creator. The features included are in-game assessment of the students' performance, adaptation of the game contents and definition of in-game guidance mechanisms.

In-Game Assessment of Students' Performance

For the in-game assessment of the students' performance WEEV is based on tools provided by the <e-Adventure> platform. In particular, it uses the "assessment report mechanism", that allows the creator of the game to define rules that will be evaluated during runtime to generate a text report with pre-defined information. The reports are generated as HTML files that can be shown to the players and sent via e-mail to the instructors. The WEEV implementation of this system hides its complexity (i.e. the rules are not created with an independent definition but as part of the narrative metaphor) while supporting advanced features such as grade-based evaluation.

The simplification of the assessment mechanism introduced in WEEV allows the instructor to immediately receive feedback about the learning outcome of a student after each game run. This feedback produces though in-game evaluation reduces the need to perform costly debriefing sessions that would otherwise be needed in most game-based learning scenarios (De Freitas*, et al.*, 2006; Squire, 2005) and increases the value of the educational video game experience (Burgos*, et al.*, 2008).

Adaptation of the Game Contents to the Users' Needs

One important aspect of video games in education is their potential to adapt the contents or challenges depending on the student's profile or abilities. This helps to keep the flow of the engagement of the player and thus increase the learning outcomes (Cordova*, et al.*, 1996; Kiili, 2005; Pablo Moreno-Ger, Burgos*, et al.*, 2009). This is achieved, for example, when a challenge is modified as to fit the current skills or knowledge of the player by making it easier or providing additional information or more detail description. This sort of adaptation is a desirable behavior in educational gaming but also difficult to implement in traditional systems (Hunicke*, et al.*, 2004).

In the WEEV system, we define a limited number of "adaptation scenarios" for the creator to choose from, each establishing a particular set of adaptation profiles (e.g. easy, normal, hard). The choice of the specific scenario is made in the wizard, but the game author can also decide not to include adaptation into the game. By contrast, in other game authoring tools specific rules and

settings must be defined for every change that the game is intended to reflect through adaptation. The new layer of abstraction introduced in WEEV makes the adaptation mechanism easier to use while keeping most of its advantages (i.e. the ability to create different game experiences for the same game).

Besides, these adaptation scenarios include the possibility to add real-time adaptation to games as a modification in the profile selected can be reflected the next time the game has to choose the most appropriate flow path. This is dependant upon an active communication with the web-based learning environment (Del Blanco*, et al.*, 2009) or specific options in the game itself.

In-Game Guidance Hints

Educational games are usually expected to cover the needs of different students. To be able to fit the needs of users with different playing styles and knowledge about the game's subject, adapting the contents is usually not enough and different adaptations must be done dynamically to the game. One of the most widely used mechanisms to achieve the needed adaptation is the introduction of a hints and guidance in the game, as some contents, such as puzzles, are particularly difficult to adapt to different knowledge levels using content adaptation.

The delivery of hints at runtime can be preformed in different ways (i.e. as text show directly on the screen or as part of a conversation with a mentor character). The definition of the hints, however, is done in a single way using the story DSVL.

6. Conclusions of this chapter

The WEEV system is presented as a methodology based on narrative concepts, design heuristics and real video game creation experience. This methodology improves upon the model used in <e-Adventure> by making it easier to use, more straightforward and tailored specifically for people with no technical background in Boolean logic or programming.

The WEEV methodology is implemented into the system as wizard that allows a structured approach to the development of video games. The wizard both guides and helps the user in the creation process, encouraging the application of good development practices (e.g. creating the virtual world before the story).

The most complex parts of the system use DSVL for their definition. The world is defined as connected spaces that the player can visit. The story uses an explicit representation of a FSM, including the representation of several educational features, and is described in detail in the following chapter.

CHAPTER V. WEEVL: WEEV LANGUAGE

This chapter provides a full description of WEEVL, the domain-specific visual language used in the WEEV system to describe the story or flow of the game. The main elements, advanced features and educational features of the language are described in detail.

1. Main elements

The WEEVL language we propose is, in essence, an explicit representation of the implicit deterministic Finite State Machine (FSM) of a video game using a state diagram. The expressive elements of the DSVL are based on the <e-Adventure> platform. WEEVL has different representation enhancements over a traditional state diagram. Some of them aim to simplify the definition of recurrent game structures that are complex to represent with a basic state diagram. Other representation enhancements are devised to add educational value using the same metaphor as the rest of the representation.

The state diagram we used, similar to a Mealy representation, has three main elements: a set of states, a transition function and an output function.

- The states used in the representation are game states. A game state is a point in the story that the player can reach and that can be associated with the possible future actions the player can perform at that point. (Figure 27, a)
- The transition function maps a state and an input (an interaction of the user with the game) to an output state. This function is represented as a set of transitions that go from one state to another (or the same one). (Figure 27, b)
- The output function maps some of the same states and inputs of the transition function to effects (see description of the <e-Adventure> platform in Chapter II). The set of effects that can be used in WEEVL is limited to those in <e-Adventure> that are perceptible to the user added to others of higher semantic value (e.g. changing the appearance of a character may require setting several variables and conditions in <e-Adventure> but in WEEVL it is done with a single atomic effect). This output function is reflected as a set of properties of the transitions. (Figure 27, c).

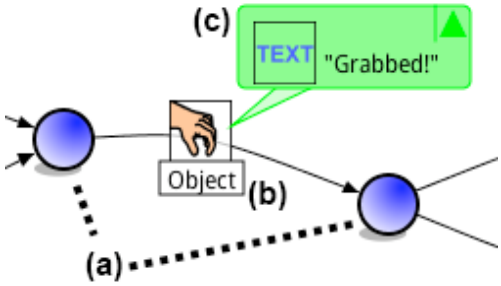


Figure 27 Two game states (a), with a transition from one to the other (b) that uses a "grab" action over the "Object" actor as an input and shows a text ("Grabbed!") on the screen as an output (c).

Another basic element of the representation is the use of a narrative structure. This concept, inspired in narrative studies, divides the story in parts that can common organizational elements in different stories. The use of a narrative structure in WEEVL allows different parts of the story to be "minimized", allowing for the edition of separate edition of the different parts while reducing the complexity of the representation (Figure 28).

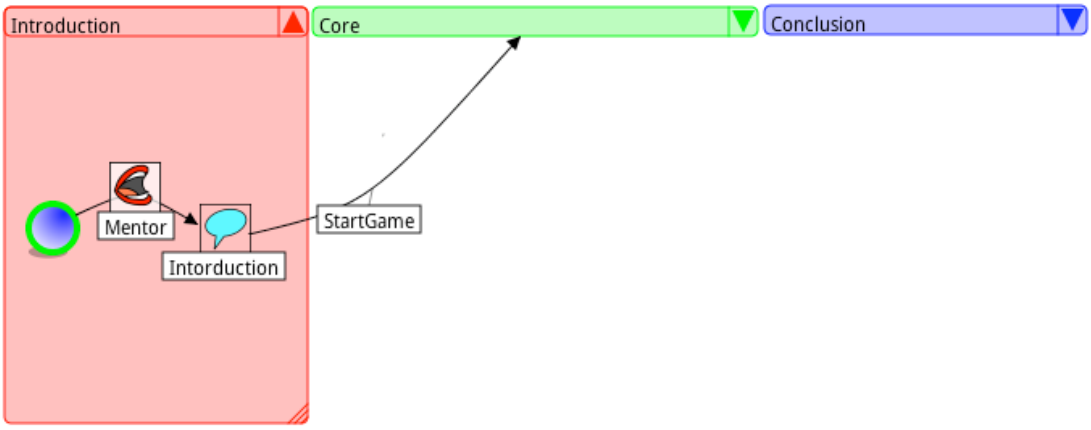


Figure 28 The story representation using WEEVL can be divided in structure parts (e.g. Introduction, core and conclusion) that can be minimized (core and conclusion in the figure are minimized) to hide complexity.

The basic narrative structure (i.e. introduction, core and conclusion) can be an effective organization for the flow of simple games. In this regard the introduction is used to present the problem in the game and provide some instructions about how to proceed and interact. Then the main actions to complete the game (and solve the problem) are developed at the core. Finally the conclusion is used to assess the acquired knowledge, present the results of the game to the student and provide feedback.

Examples of the use of the basic narrative structure to organize the story-flow in WEEVL can be found in the case study presented later in this work. Nevertheless more complex narrative structures can be used to group the elements of longer stories.

Node icons

Figure 29 Key to the default nodes in WEEVL, it does not include the icons in the representation enhancements

The WEEVL representation uses one basic kind of node that represents the current game state of the player (“Regular state”) and in each story one of these nodes is marked as the initial game state (“Initial state”) (Figure 29).

The actual representation of games uses more node representations, such as that used in conversations or to represent randomness, but these nodes are introduced as representation enhancements.

Transition icons

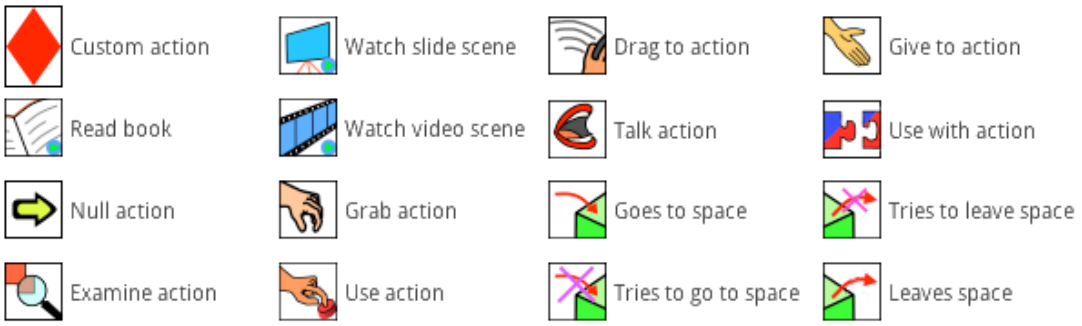


Figure 30 Key to all the default transition icons used in the WEEVL language. These icons are in no particular order and do not include the special icons used by representation enhancements

The icons used for the transitions represent actions the player takes in the game (Figure 30). For instance "Use with action", "Drag to action", "Talk action", "Give to action", "Grab action", "Examine action", "Use action" and "Custom action" represent all the basic actions available in <e-Adventure>.

Other icons represent actions of higher semantic value (represented by several different relationships and conditions in <e-Adventure>) such as "Goes to space", "Tries to leave space", "Tries to go to space" and "Leaves space", and represent different consequences of the player moving though the world.

Finally, some icons describe passive actions in <e-Adventure>. These actions require no active involvement from the player (once a node with one of this actions is reach, it will be taken without intervention from the player). They are represented by "Watch video", "Watch slide scene", "Read book" and "Null action" (this last action just skips to another node).

Effect icons

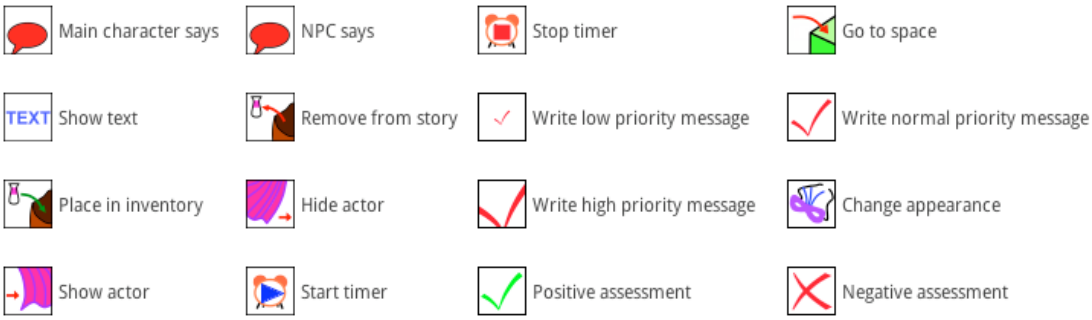


Figure 31 Key to all the default effect icons used in the WEEVL language. These icons are in no particular order.

Effects in WEEVL represent either equivalent effects in the underlying <e-Adventure> model or have higher semantic value (Figure 31). "Main character says", "NPC says", "Show text", etc. are some of the effects that have a direct equivalent in <e-Adventure> and as such can be directly converted.

However, other effects have a higher semantic value that requires for their transformation several conditions to be met. For instance, "Start timer" and

"Stop timer" do not exist as such in <e-Adventure>, but can be created by modifying a flag linked to the starting of the timer. The same is true for effects such as "Show actor", "Hide actor" and "Change appearance". Other effects, such as those linked to evaluation ("Write … priority message" or "Positive assessment") need special transformations, including the creation of the necessary evaluation rules in the evaluation profile for the game.

2. Representation enhancements

Previous experience with the use of FSM or other graphic representations to describe games (Pablo Moreno-Ger, Fernández*, et al.*, 2009) shows that it may become overly complicated to represent a full game in a way that humans can directly understand. To tackle this problem we use a system of representation enhancements that are reflected in the visual language with different graphic elements that have higher semantic value. We consider two types of representation enhancements: the first tries to make the story-flow designs more homogeneous and readable while the second aims to simplify the visual representation of complex concepts.

The first group of representation enhancements includes the possibility to represent passive actions (e.g. watching a cut scene) in the same way as active actions (e.g. grabbing an object) even though they are not treated in the same way in the underlying <e-Adventure> representation (Figure 32). This allows the representation of the story-flow to be complete (every possible action by the user can be represented) and homogenous (every action the user can perform is represented in the same manner).

Figure 32 When the story reaches a node with a passive action, this will automatically start. This figure shows that the action of watching a cut scene (named "CutScene") will take the story from the state on the left to the one on the right.

Other enhancements are meant to reduce the perceived complexity of the representation. For example, the representation hides the underlying <e-Adventure> variables and conditions that represent the game state. WEEVL treats effects and changes in the game with abstractions built on top of the variable/condition system, yielding a higher semantic value and explicit meaning, thus making the representation easier to understand.

Specific enhancements are provided to support use patterns that frequently arise in games or situations that would require an increased number of transitions with the basic state-transition representation. These enhancements allow for the definition of out of order sequences, random situations, parallel story lines and other patterns. Each enhancement has its own particular representation.

Each relevant representation enhancement will be dealt with in detail in the following sections.

Conversations

The conversation representation allows the description of any possible conversation in the underlying <e-Adventure> language. Conversations can have several lines of text, belonging to the main character or player and the other actors in the game (only NPCs in current versions, given the limitations in <e-Adventure>).

Conversations can have one transition from them, in which case that transition will always be taken (i.e. no choice will be presented to the user). When a conversation has several transitions, the player will be shown the choice and the narrative will change accordingly (Figure 33). In conversations, the transitions show the text of the option given to the player and no icon.

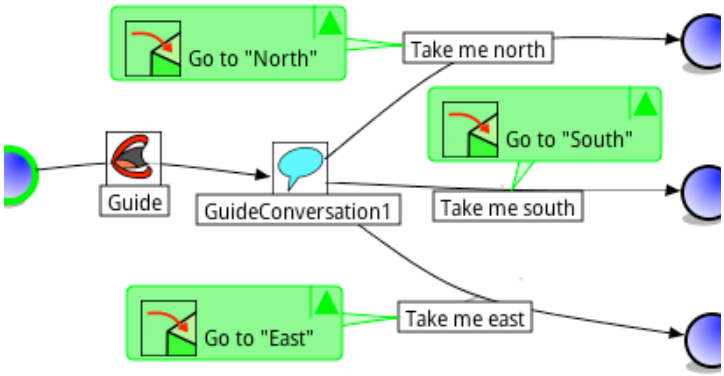


Figure 33 In a conversation (GuideConversation1) the player has 3 choices ("Take me north", "Take me south" and "Take me east"), each with resulting in a different effect and a different path for the story

Multiple-choice conversations

Figure 34 Multiple-choice transition icons

The multiple-choice conversation representation allows the straightforward definition of conversations that end with one correct response and one or several incorrect ones. This representation makes it simple to represent a particular conversation that is common in adventure games, but the same behavior could be achieved using the regular conversation representation, albeit with more “unnecessary” transitions. Multiple-choice conversation use special transition icons for the correct and incorrect answers (Figure 34). When a multiple-choice question is presented to the user (Figure 35, a), the correct answer will have one consequence in the story (Figure 35, b) and the incorrect answer will have other consequences (Figure 35, c).

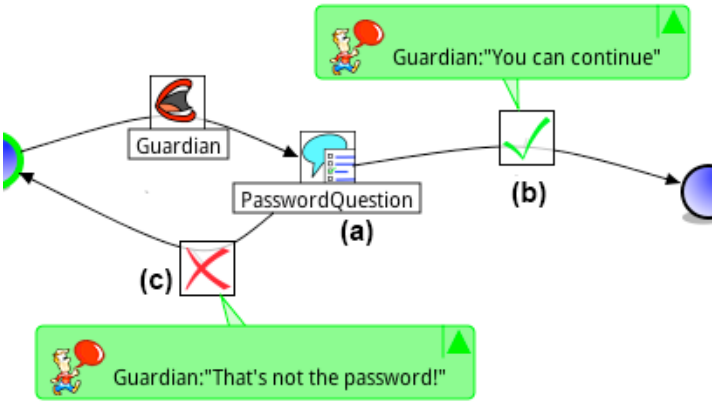


Figure 35 Multiple-choice questions, such as on that asks for a password (a), can have different consequences when answered correctly (b) or incorrectly (c)

Multi-interactions

A specific enhancement is provided to support for out of order sequences. It receives the name of "Multi-interaction" in the WEEVL system, and it allows complex behaviors to be expressed in a simplified graphic element.

"Multi-interactions" allow for the representation of both simple and complex behaviors. Simple out of order behaviors allow the player the possibility to perform some actions any order as long as every one of them is performed. For example, the player might have to grab two objects (salt and pepper) from a table, but the effects on the story will not change if he/she grabs the salt and then the pepper or the pepper and then the salt (Figure 36). In the representation of the example in the figure, once the player grabs the salt and pepper, a cut scene will start playing.

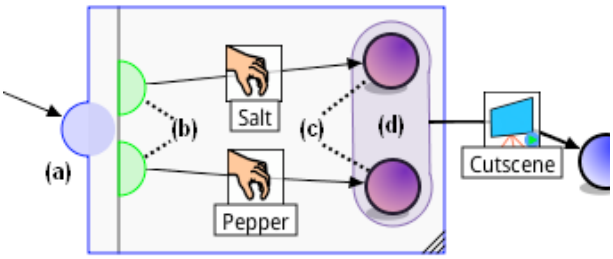


Figure 36 A simple use of a "multi-interaction" (a) using two start nodes (b), two end nodes (c) and a node group (d) to express the out of order sequencing of two actions ("grab salt" and "grab pepper") followed by a cut scene.

The "multi-interaction" representation uses start nodes (Figure 36, a) to represent the different starting nodes of the possible actions available to the player. The end nodes (Figure 36, d) represent final states the different actions (or sequences of actions) in the "multi-interaction" can reach. The node group (Figure 36, c) is the set of nodes that must be reached for the story to continue the flow from the transitions that start from it.

"Multi-interactions" can have multiple end node groups to represent different states the player can reach in the game that allow particular actions with different consequences. An example of such a representation can be used to

describe, for instance, the recipes for different kinds of chocolate (P; Moreno-Ger, *et al.*, 2007) (Figure 37).

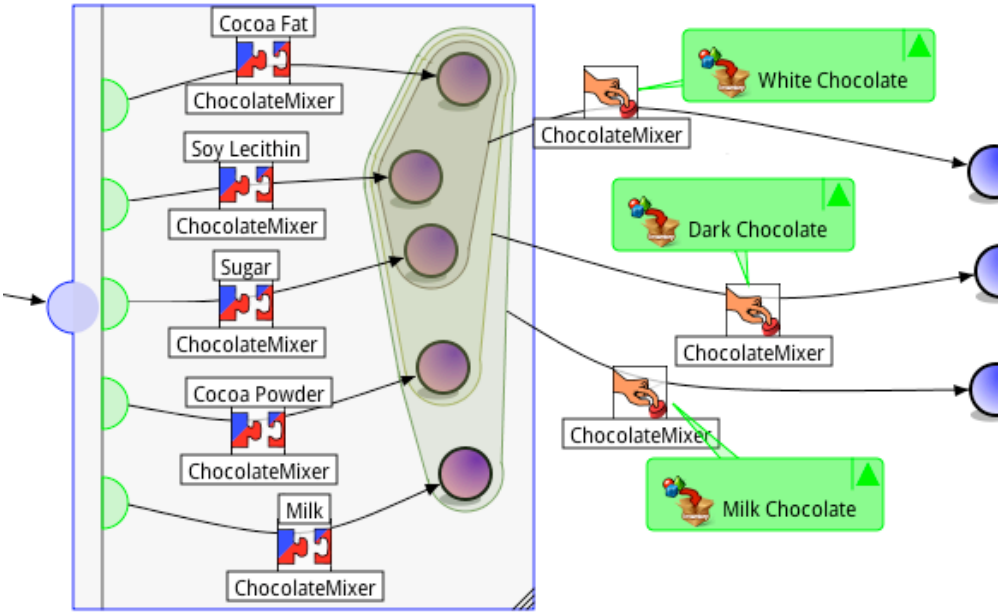


Figure 37 "Multi-interactions" can be used to describe complex procedures, such as the mixing of ingredients to create different kinds of chocolate in a chocolate mixer.

Three kinds of chocolate can be mixed in the example: “White chocolate”, “Dark chocolate” and “Milk chocolate”. The different ingredients of the mix are placed inside a “Chocolate Mixer”, and when the correct combination is placed the appropriate chocolate will be created. In the example, “Cocoa fat”, “Soy Lecithin” and “Sugar” are mixed to produce “White chocolate”.

Random nodes

In order to describe complex situations in the story-flow that would require increased number of transitions (e.g. timers or random effects) we introduce another representation enhancements. For example, special random states are used to include random behavior in the game (Figure 38). These nodes have one “arrive” node (Figure 38, a) and two “leave” nodes (Figure 38, b) with different probabilities. If the story reaches the “arrive” node, it the will continue from one of the “leave” nodes depending on their probability.

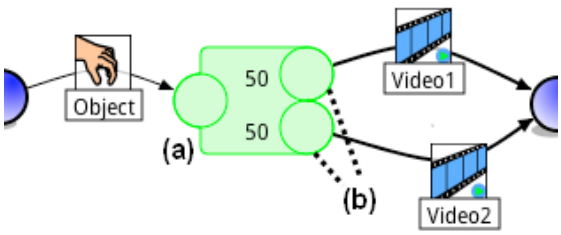


Figure 38 Using a random state, after the player grabs the object and reaches the “arrive” node (a) the story will continue in either “leave” node (b) with a 50% chance for either one, resulting in Video1 or Video2 beeing shown.

Timers

Timers, through the timer representation, allow the definition of time-sensitive behaviors of the game within the graphic representation. Timers use the same state machine representation, with the timer being the initial node of a flow that will start when a configurable time expires. Timers can be started and stopped using the effect system in the transitions.

Figure 39 shows how a timer can be used to introduce a “time dependant” behavior in a game. In the interactions described, the player will activate a bomb by grabbing it, and if it is not deactivated within 20 seconds a video of an explosion will be shown.

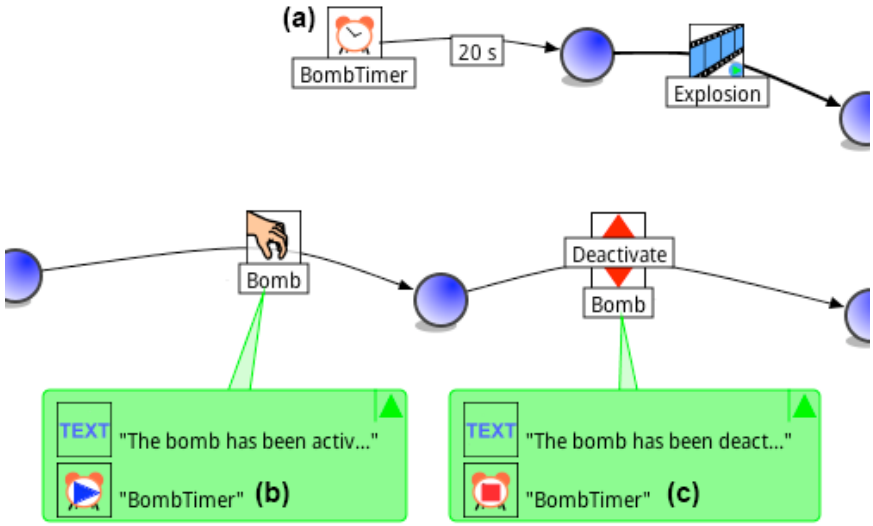


Figure 39 Timers can be used to introduce "time dependant" transitions. Once the "Bomb" is grabbed, the "BombTimer" (a) is started by the effect (b). When the "Bomb" is deactivated, it will be stopped by the effect (c). If it isn't stopped within 20 seconds, a video of an explosion is shown

Parallel story lines

Parallel story lines allow the definition of flows that do not affect the main story. The player can start a parallel story at different times, indicated by a special transition from the nodes where this is possible. The player is able to advance in the parallel flow at the same time than in the parallel story.

This representation is useful when a series of interdependent actions are available for the player at different times. For example, when the player can grab a book, examine it and read it (Figure 40, a) at different point in the story this representation to which the parallel story line is connected (Figure 40, b).

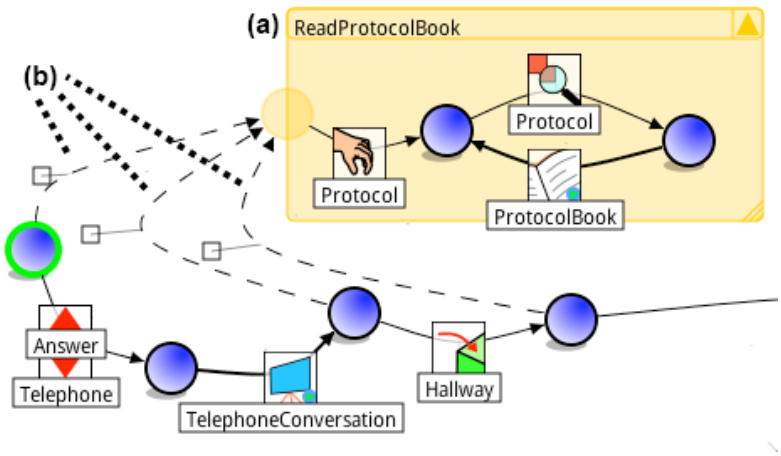


Figure 40 Parallel story lines (a) are connected to different nodes from which they can start using paralell story line transitions (b)

Virtual states

Virtual states are similar to parallel stories in the sense that they make it possible to define a series of transitions common to several nodes. However, transitions that start from a virtual state modify the flow to which the state is connected.

A typical use of virtual states is to use them when a certain action has nefarious consequences (i.e. the game ends). In other cases, virtual state can help define simple actions that have no relevant consequences. In the example in Figure 41 the player has to use three switches in order, but can try to open a door at any time without consequence until the last switch is used in the correct order.

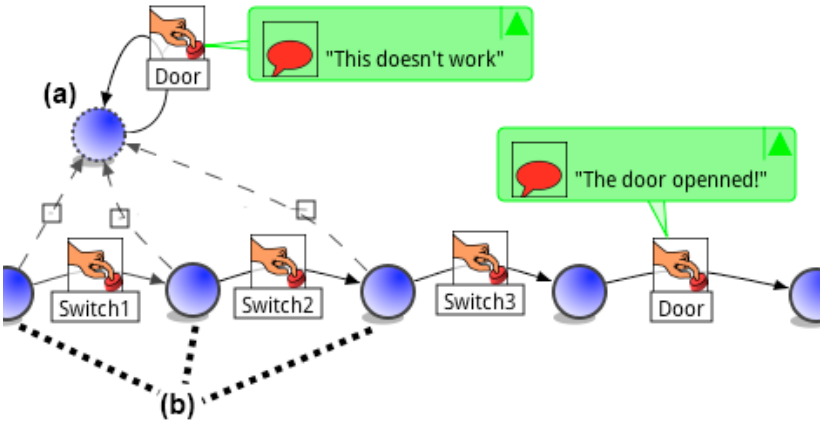


Figure 41 A virtual state used to define the same action for several nodes

3. Story-parts

The complexity of the story representation using WEEVL can increase when the story becomes bigger. In such cases it is required that the language includes mechanisms to encapsulate parts of the story, allowing the designer to think about the story at different levels of detail. To achieve this, an element called "story-parts" is introduced to obtain a clearer story-flow representation (Figure 42). These elements are similar in functionality to a function or method in a general-purpose programming language, encapsulating groups of steps that will have clear consequences into a single element. This kind of representation is identified as a "hierarquical representation" within DSVL theory.

When a "story-part" is introduced, it appears as a special kind of state in the representation with a name (e.g. "challenge") and will have labeled transitions to other states (e.g. "challenge mastered" or "challenge failed").

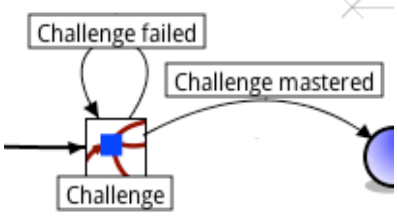


Figure 42 "Story parts" allow the encapulation of sections of the story-flow that have clear consequences. In this case the "Challenge" story part can result in either the "Challenge mastered" in which case the story continues or the "Challenged failed" that will force the player to retry.

The story part is then defined separately inside the same editor, using all the elements available in WEEVL. The story part will have a single entry state and a different end state for each one of its labeled transitions in the story-flow representation (Figure 43).

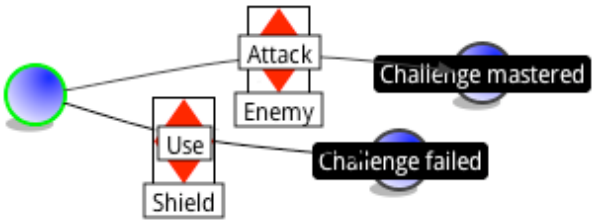


Figure 43 Editing inside the "story-part" is done as with the rest of WEEVL. In this story part, the player will master the challenge by attacking the enemy and fail it if trying to use the shield.

4. Representation of educational features

As the <e-Adventure> platform is specially oriented to education, it also provides education-specific features (Pablo Moreno-Ger, Burgos, *et al.*, 2008; J. Torrente, *et al.*, 2008). WEEVL incorporates these features through the use of special representation elements. Two of the foremost educational features are the ability to adapt the game to the user needs (i.e. adaptation) and the capacity of the system to inform the instructor or teacher of the progress made by the users of the game (i.e. evaluation).

Adaptation of the game flow

Adaptation is represented in the story-flow through the definition of alternative flow-paths inside the different "story-part" elements. The user will experience a game run just as described by the path that better fits his/her current needs. Two adaptation mechanisms are supported: on-the-fly adaptation and initial adaptation. For on-the-fly adaptation the alternative paths fork at different points in the game automatically allowing the dynamic adaptation of the game (e.g. if the game is being too easy, the next time it can fork to a harder path). For initial adaptation the game asks for some parts of the user data model stored in a learning environment and depending on its values decides the initial game state. Besides, as this representation just needs the creation of different story-flows it uses the same metaphor of the rest of the system.

If an "adaptation scenario" was selected in the wizard, when the time comes to define the story flow of the game the user is encouraged to define in certain sections of the description how the game should continue. For each adaptation profile in the scenario, the story will have a different start node in each story-part. Using this information, the most appropriate flow for the story is decided by the game in real-time. An example of this use is presented in Figure 44, where the game author determines that a different video is presented to the player depending on whether the game is set to the easy, medium or hard profile.

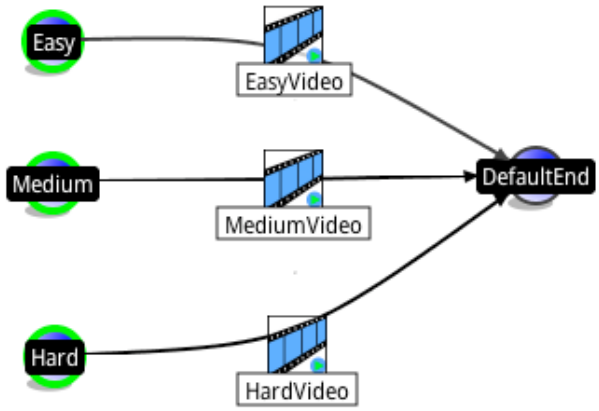


Figure 44 View of the edition of a story-part when a "Difficulty" adaptation profile is selected to show different videos for each difficulty

Evaluation or assessment of in-game student performance

Evaluation, assessment or tracking of the user interaction within the system is preformed through the use of effects. This again is a high-level abstraction of the underlying system that depends on variables and conditions to assess the performance. There are two kinds of effects available, one of them just writes a log line in the assessment report shown at the end of the chapter and the other allows the user to define a variation in the global score of the user. This system is an explicit representation of the proposal by Moreno-Ger (2007) to represent the internal evaluation as preformed by <e-Adventure> games.

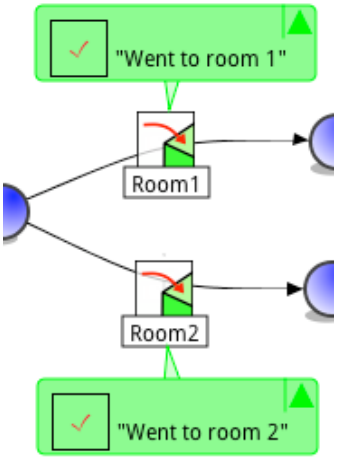


Figure 45 The player can choose to go to room 1 or 2, and a low priority line will be written in the assessment report, with no consequence to the global score

"Write report" effects just write information in the assessment report. This can be useful, for example, to document a choice made by the player that does not affect the score (Figure 45). These effects can establish different priorities for the lines, as established in <e-Adventure>, which allows the player to generate assessment reports with different levels of detail.

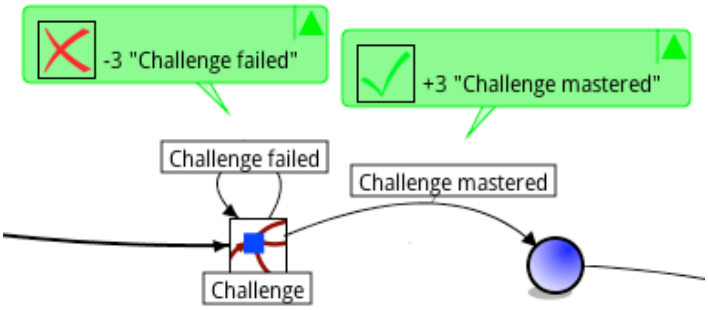


Figure 46 Evaluation effects can assess possitive or negative actions by the user. In this case, a possitive assessment is created when the player masters the challenge in the story part and a negative assessment is created when the player fails the challenge.

"Assessment effects", besides writing in the assessment report modify a "global score" variable. The user can set how this variable is modified, either increasing or decreasing its value in 1 to 5 points. The effect then appears in

the representation clearly indicating if the evaluation is positive or negative (Figure 46). This allows a simple result to be returned from each game run. This value is especially useful for games that are integrated in a web-based learning environment that supports global score feedback from the contents (such as that supported by the SCORM specification).

In-game guidance hints

In-game guidance hints (i.e. help) are represented as properties of nodes. This allows different hints to be shown depending on what the user can/must do at different game states. The hints are represented as properties of states (Figure 47).

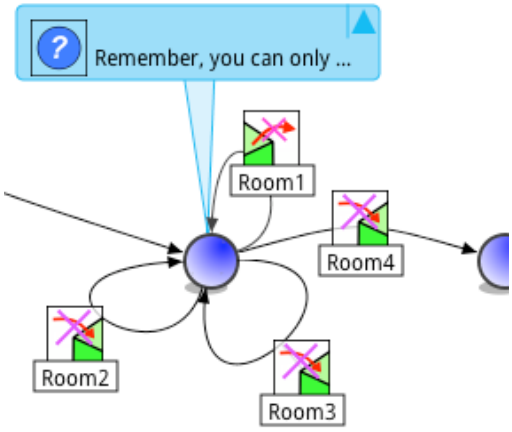


Figure 47 Hints show as properties of nodes (a) and can help the player along the game

During game-play, a button will appear on-screen if hints are available in the game. When the user clicks this button, the text of the hint will be shown. At the same time, every time the user asks for a hint the system will log this in the assessment report and will reduce the value "global score". This is intended to discourage the use of hints so that players still try to explore the game world and the possibilities, even if they still can fall back on the hint system when necessary.

5. Conclusions of this chapter

This chapter shows that the WEEV language (WEEVL) is a DSVL (or VPL) that includes several elements to represent games, to enhance the representation and to include educational features in the same metaphor. The elements presented in this chapter are fully implemented in the system and allow the full description of the stories in educational video games.

The representation enhancements included provide assistance in the creation of structures that are repeated across different educational video games (e.g. "multi-interaction" nodes) and, some of them, video games in general. This allows the representation of complex behaviors in a straightforward manner that is helpful both to create the game and understand a story of existing games.

The representational elements aimed at reducing the complexity (i.e. "story-parts") allow great flexibility both to describe complex situations within the same metaphor and to reuse parts of the story. Besides, the use of this representation within the adaptation framework allows easy to understand adaptation of games while keeping the details within limits to improve the comprehension of the story as a whole.

Finally, the representation of educational elements allows the inclusion of otherwise complex educational features such as evaluation and in-game hints within the same system and using the same metaphors. By doing this, the representation allows complex educational games to be more useful both to educators that can easily establish the educational outcomes and for students that can use the hint system to progress in the game even if they find parts of it to difficult at the time.

CHAPTER VI. THE WEEV IMPLEMENTATION

This chapter includes a description of different aspects of the implementation of the WEEV system. A general description of the implementation, the process by which the description of games is transformed to playable <e-Adventure> games and the extensible architecture that allows new components to be easily added are provided.

1. Java implementation

The WEEV system was implemented using *Java SE 6.0*. The implementation follows a Model-View-Controller (MVC) design patter (Figure 48) while reusing the already existing <e-Adventure> code when necessary. The current version consists of more than 22.000 lines of code, 9.000 lines of comments and 300 classes, representing around 17% of all the code in the <e-Adventure> project where it is included.

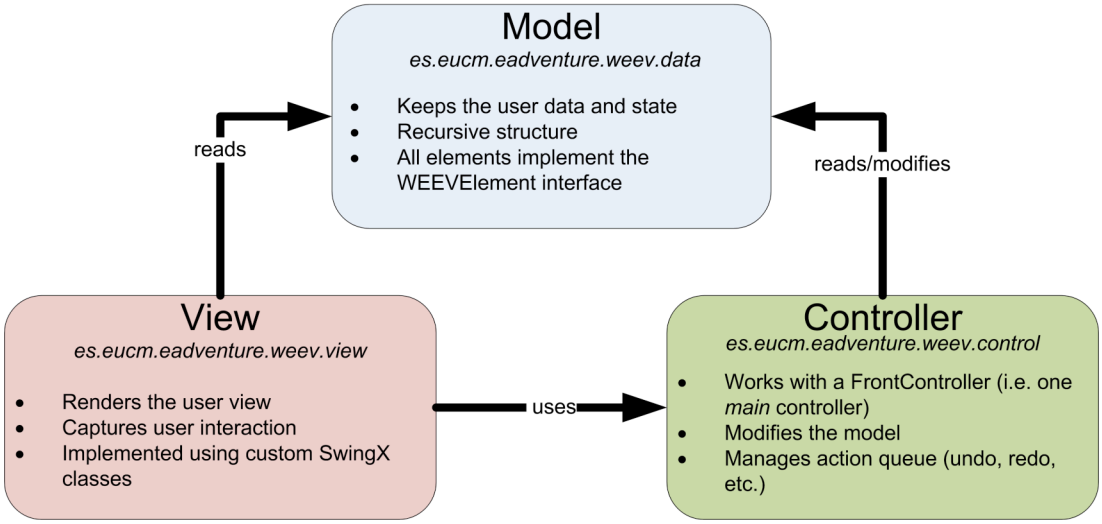


Figure 48 High-level overview of the MVC as implemented in the WEEV system

Visual elements (including both the edition panels and the buttons, combo boxes and other elements) were recreated from scratch, in an effort to create a coherent, consistent, easy to use and visually appealing interface. The Java Swing element library developed is being used as the component library for the next version of the <e-Adventure> platform as a whole, as it provides better multi-platform compatibility and a unique look to the whole application environment.

2. An extensible architecture

The WEEV implementation was designed to be extensible. The current version does not directly support the dynamic inclusion of plug-ins, but the representation, data and control elements (e.g. transformation into <e-Adventure> games) of the system were created to support such functionality. The current "statically extensible" implementation allows the incorporation of new representation elements and tools without modifications to the main code.

The aims of such software architecture are clear: make the addition of new components as easy and straightforward as possible. This has enabled the growth in representation elements with little or no effort, while keeping the code correctly encapsulated and allowing independent testing of different sections.

Currently, just a few Java interfaces need to be implemented or abstract classes extended by the programmer to add a new representation element in the story flow. For instance, the classes that must be implemented to create a new kind of node (all the classes belong to the *es.eucm.eadventure.weev* package):

- *...data,story.elements.Node*: This abstract class must be extended by any element that is used as a node in the representation.
- *...data.story.elements.Transition*: This class might need to be extended if the new node needs special transitions.
- *...view.story.StoryElementView*: This abstract class needs to be extended indicating how the element is represented in the story panel, as well as the right click behavior.
- *...view.story.StoryGraphTool*: This class needs to be extended to add a tool that will allow the new kind of node to be added to the representation. This tool defines the icon in the toolbar, the cursor, etc.
- *...control.converter.NodeProcessor*: This class needs to be extended to implement the conversion of the representation element to the intermediate model used by WEEV during conversion.
- *...control.converter.TransitionProcessor*: This class might need to be extended if the new element has new transitions that need to be converted to the intermediate model.
- *...control.tools.Tool*: This abstract class might need to be extended if the new element needs special modifications to the model (it isn't needed to add the new node to the model, for instance).

As all elements are created this way, the programmer might find inspiration in some of the default elements while adding a new one to the representation. It must be noted that in current versions the *StoryElementView* implementation, the *StoryGraphTool* and the processors (*NodeProcessor*, *TransitionProcessor*) must be registered with their corresponding Factories (e.g. *...view.story.StoryViewFactory* for *StoryElementView*).

3. Transformation of games

The transformation of games into <e-Adventure> games presents different challenges. Firstly, <e-Adventure> is not strictly a programming language and does not support scripting, so the final representation is an object-oriented representation of an <e-Adventure> game. These games can be directly edited in the <e-Adventure> editor. Secondly, the plug-in or extensible architecture used by the system requires the use of an intermediate language to reduce the complexity of the implementation of such extensions and achieve the needed modular design.

However, the story-flow created using WEEVL, along with every other aspect of a game defined in WEEV, can be directly converted into the <e-Adventure> video game it describes. This transformation is automatic and transparent to the user, who can continue editing the game in <e-Adventure>. What is more, changes to graphic resources, placements, etc. are directly reflected in the WEEV interface and saved with the WEEV project.

The two-stage process uses an intermediate language, which can be described as an annotated Finite State Machine (FSM). By visiting the elements of the WEEVL description and applying algorithms defined specifically for each representational element the representation using the intermediate language is generated. The intermediate language uses no representation enhancements, story parts, etc. and can be directly converted to the <e-Adventure> game. This <e-Adventure> game can then be executed in the game engine (Figure 49)

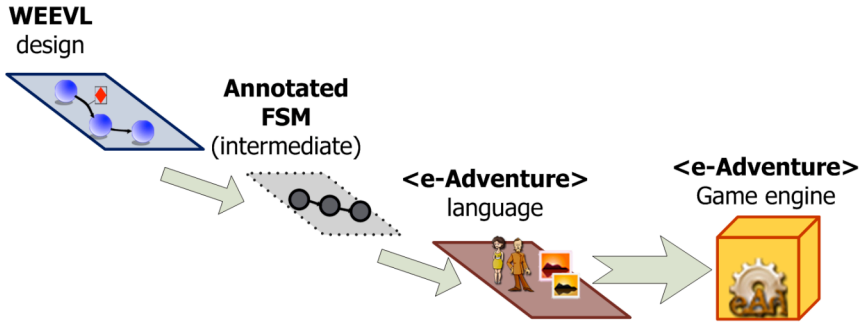


Figure 49 The system uses an intermediate step for the transformation. In this intermediate step an annotated FSM is generated that is later converted to an <e-Adventure> game

4. Other implementation considerations

Different elements in the interface are susceptible to different uses. Details can be edited for all elements in the system (e.g. spaces), but these options are only accessed using the contextual menus (accessed through the right mouse button). When details are configured, two options to compile the game are presented: create a game to be edited further in <e-Adventure> proper or run the game directly.

Editing element's detailed information

Most elements in the WEEV system are only described by their function, as opposed to <e-Adventure> where there are described mostly by their graphic resources. This description, however, is accessible inside the WEEV system and can be configured without the need to use the <e-Adventure> editor.

To access the detailed information of elements, users must right-click the different elements and select the appropriate choice. For example, "Change appearance" of a space will allow the user to set the background image, the foreground mask and the sound of the equivalent <e-Adventure> scene (Figure 50).

The option "Place actors" of the scenes, for instance, will allow the edition of the placement of actors in the scene directly within the WEEV interface. This way, the user does not need to use the <e-Adventure> editor.

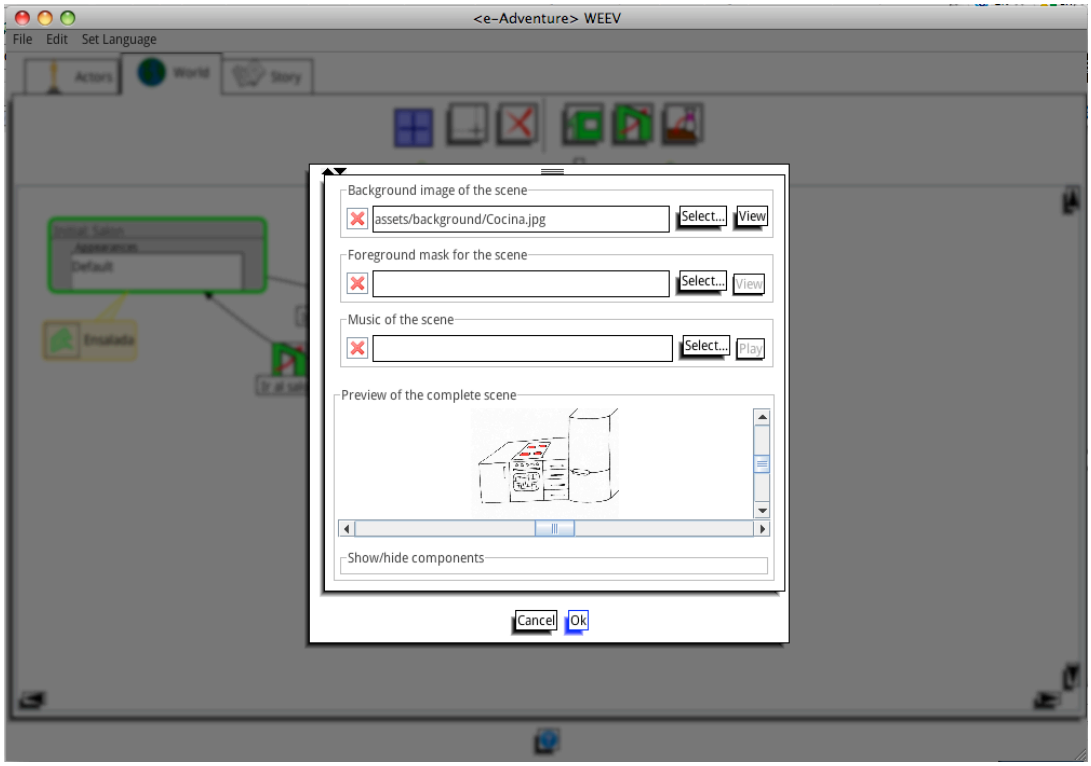


Figure 50 The graphic resources of all elements can be directly edited whitin WEEV. In this case, the background of a space is being edited

This same behavior is available for different elements in the story representation. For instance, choosing the "Edit conversation" option for a multiple-choice conversation will allow the user to edit its details (Figure 51).

In the case of multiple-choice conversation the user can edit the conversation (Figure 51, a), and the correct answer and wrong choices (Figure 51, b).

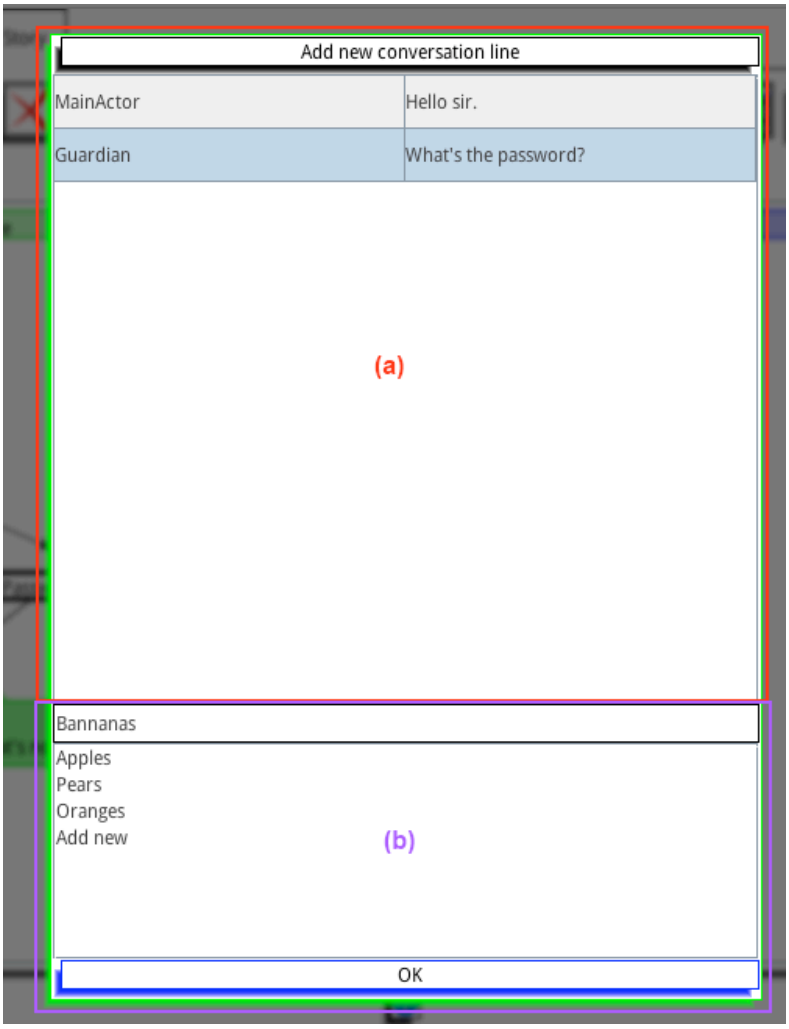


Figure 51 The WEEV interface allows the edition of details directly within WEEV. In this figure, the conversation (a) and the answers (b) in a multiple-choice conversation can be edited

Converting and/or running a game

The WEEV interface allows to different uses of the system. On the one hand, the user can choose to convert ("File" -> "Convert") the game directly to a <e-Adventure> game and continue editing the details in the <e-Adventure> editor. On the other hand, the user can choose to run the game directly ("File" -> "Run") as long as the details (i.e. graphic resources) are correctly configured for the game. This too approaches fit with the definitions of "purely visual language" and "hybrid text and visual system", respectably, provided by Boshernistan *et al.* (2004).

Figure 52 shows how a complete educational game, with all its details configured, can run directly from within the WEEV interface. The game show in the screen-shot is described in the use cases (The salad game: A simple educational example).

Figure 52 A complete education game running directly from the WEEV interface, the <e-Adventure> editor was not needed to try the game

5. Conclusions of this chapter

This chapter details different aspects of the implementation of the system to provide a glance at the potential of the system to be used in real contexts and extended further.

The current implementation is based on a transformation of WEEV games into <e-Adventure> games, described in this chapter, which is flexible enough to support the extensible architecture also described in some detail. Besides, the use of software engineering design patters helps to achieve a better and more flexible code.

The system extends the actual WEEV metaphor and methodology by allowing implementation details such as graphic assets to be defined within the same interface, by reusing components of <e-Adventure>. This opens the possibility of games being tested directly from within the WEEV interface, allowing users to skip the use of the traditional <e-Adventure> editor in many cases. However, if users choose to use the <e-Adventure> editor, changes to graphic resources will be updated in the WEEV system, allowing users familiar with the traditional editor to continue using the old interface.

CHAPTER VII. USE CASES

This chapter presents the full implementation of some games using WEEV. These games include the simple game used in the evaluation of the platform, the others are available at the <e-Adventure> website. One of them was even used in a real educational context.

1. The salad game: A simple educational example

"The salad game" is a simple game proposal used to introduce new users to the platform. In this game, the player will learn how to dress a salad with olive oil, vinegar and salt. The player will be forced by the system to use the ingredients in the correct order (salt and vinegar in any order first and the olive oil) according to Spanish "popular knowledge". The game evaluates the user positively or negatively depending on how he tries to dress the salad.

The resources for this game are included in the standard WEEV distribution and the creation of this game is described in the tutorial and the guided experience used in evaluations of the system.

This game is a first person or interactive world game. It uses no adaptation by default and no story structure (it is to simple a game to benefit from it).

World

The world in this game is rather simple, including only two spaces. The player can go from one place to the other. The salad is placed in the "Dinning room" space and the ingredients (i.e. salt, vinegar and olive oil) are in the "Kitchen" space (Figure 54).

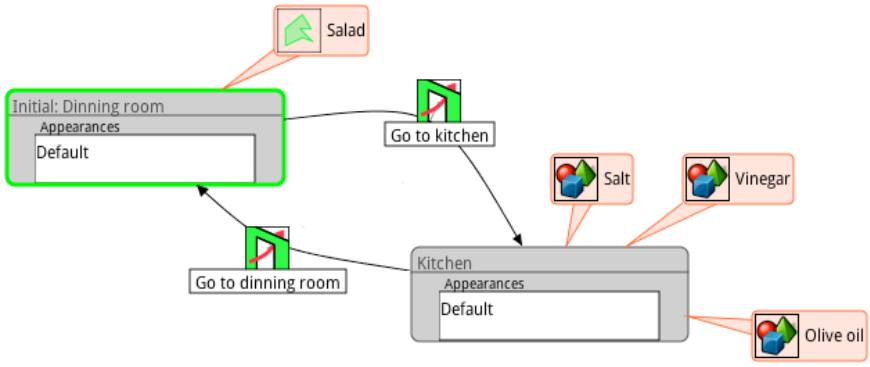


Figure 53 The world in the salad game example

Story

The story includes feedback and evaluation, as well as hints (Figure 54). The player will start in the "dinning room", where he/she will find that the salad placed over the table hasn't been dressed. The next step is going to the kitchen to find the necessary ingredients. Once the ingredients are in the user inventory, he/she will proceed to the "dinning room" once more and dress the salad. The salt and vinegar can be used in any order (represented by the use of a "multi-interaction") while the olive oil must be used last.

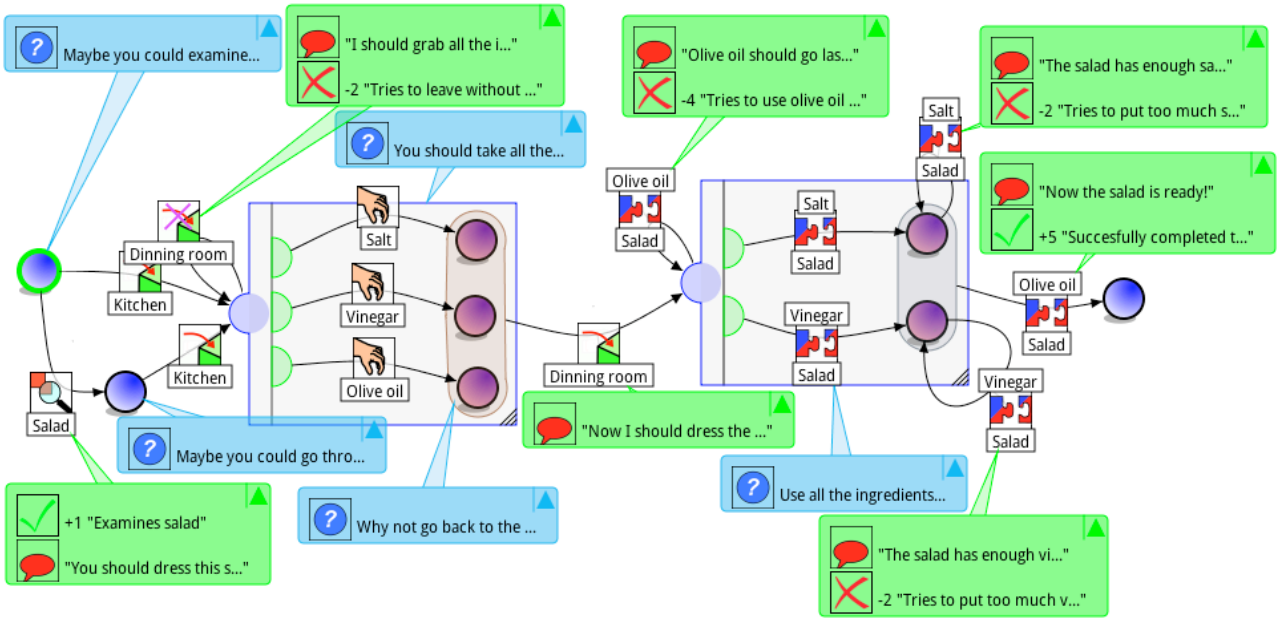


Figure 54 The story view of the salad example game

Playing the game

The game can be played directly from the WEEV system (Figure 55). The hint button is automatically included into the game so the user can access the hints when available.

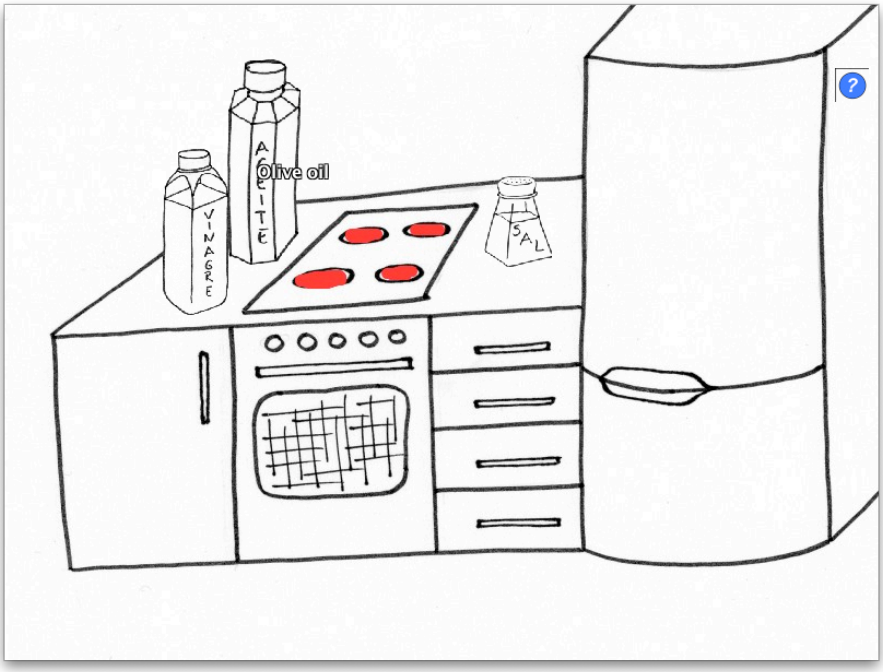


Figure 55 The salad game during game-play

2. The fire protocol game

The fire protocol game was originally created as an example of the types of games that could be created using the <e-Adventure> platform. This game involves the protocol that must be followed to evacuate a building (the Computer Science School in particular).

World

The game takes place in the Computer Science School, involving some of the offices there and the halls through with the player moves. There are different elements with which the player can interact along the way, as well as one office that is on fire (Figure 56).

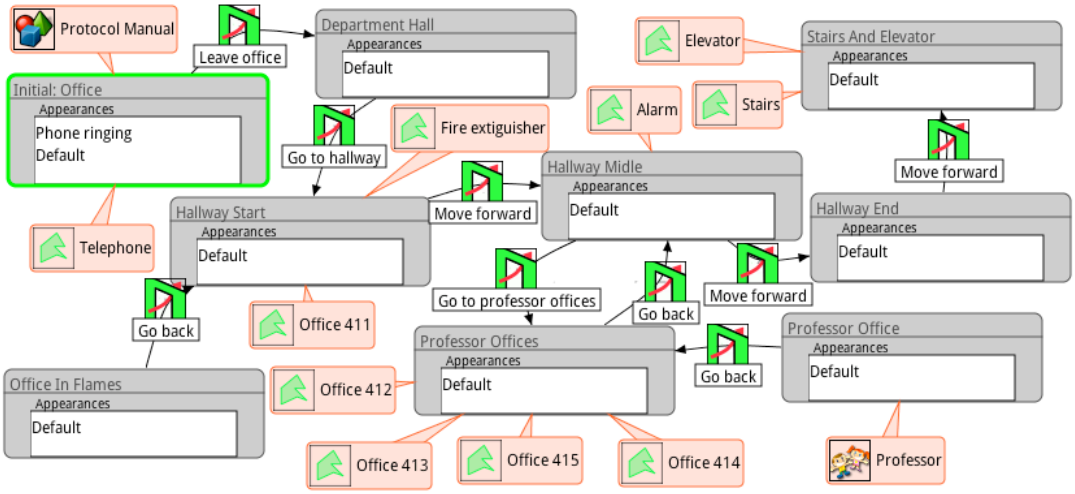


Figure 56 The world in the fire protocol game as represented in WEEV

Story

The story in the protocol game involves the evacuation of the School building following the fire evacuation protocol. The game starts when the player receives a call informing of a possible fire in one of the offices. The steps that make up the game are:

1. Answer the telephone to get information about the possible fire
2. Check the office that is supposed to be in flames
3. Activate the fire alarm
4. Check other offices in the floor to make sure no one is left behind
5. Leave the building, using the stairs

Along the way, different choices are available to the player, such as mistakes that can be made that have important consequences (e.g. using the elevator ends the game badly) or insignificant ones (e.g. trying to use the fire extinguisher).

During the story, the player is evaluated to establish how well the protocol is followed. The information about the player's performance is used to generate a report. The story is implemented by the use of story-parts to establish the

details of the most "complex" interactions, leaving the main storyline as simple as possible (Figure 57).

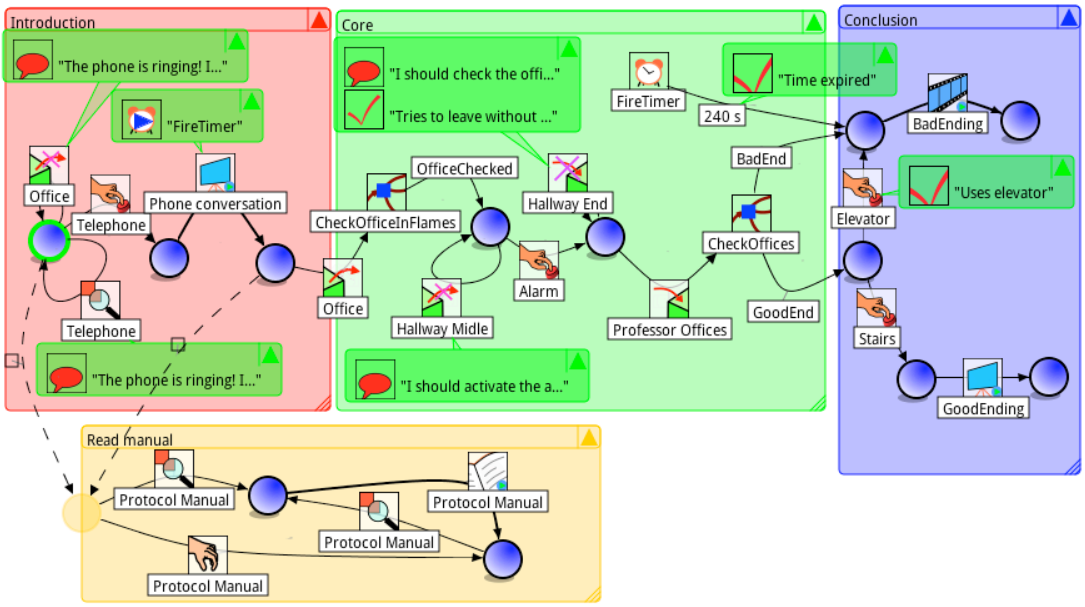


Figure 57 The main storyline in the fire protocol game

Story-part: CheckOfficeInFlames

This story-part details the interactions the user must perform to check the office that is supposed to be in flames (Figure 58). Besides, it also includes some choices for the player to make such as the use of the extinguisher both before and after checking the office.

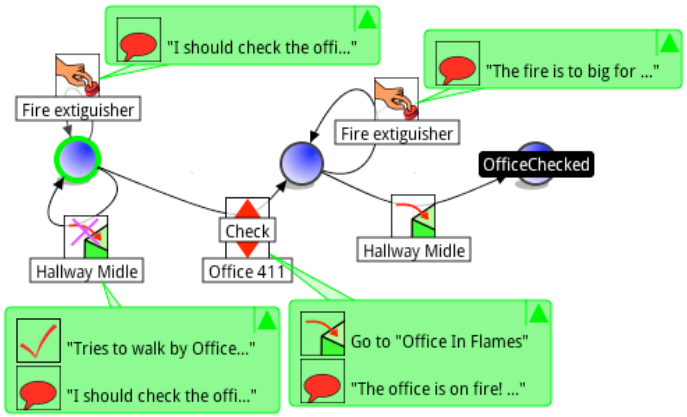


Figure 58 The "CheckOfficeInFlames" story-part in the fire protocol game

Story-part: CheckOffices

This story-part details the interactions required of the player to check the offices in the floor to make sure no one is left behind (Figure 59). The player must check the four offices that are to be found in the way. The order is not relevant, so a multi-interaction node is used. The player will find one of the offices to be occupied, and will have to convince the professor there to leave the building. This is conversation is detailed in another story-part.

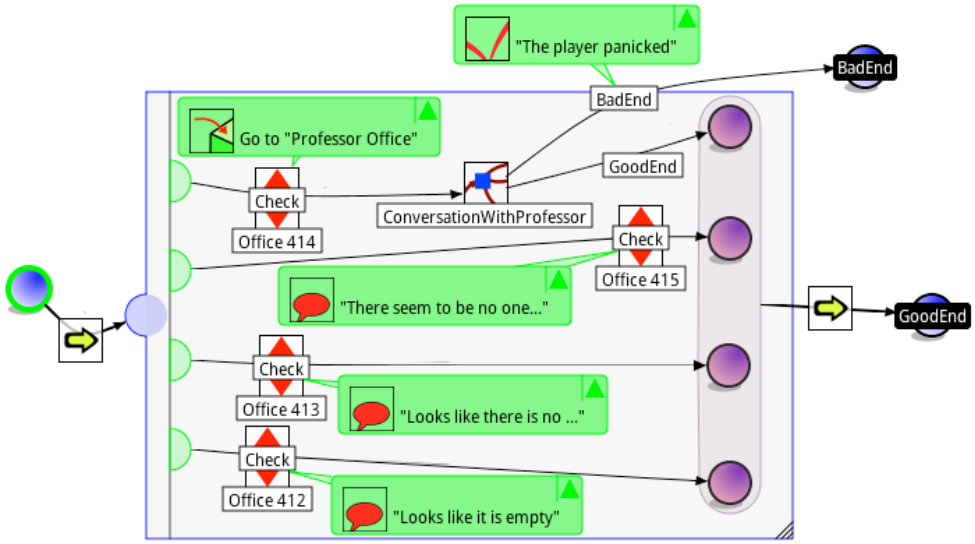


Figure 59 The "CheckOffices" story-part in the fire protocol game

Story-part: ConversationWithProfessor

This story-part details the conversation the player has with the professor that is inside one of the offices (Figure 60). This conversation can lead to bad endings, mainly as a consequence of the player panicking. In any other case, the conversation will be successful even if some of the choices will be written in the report as they are not behavior that should be encouraged.

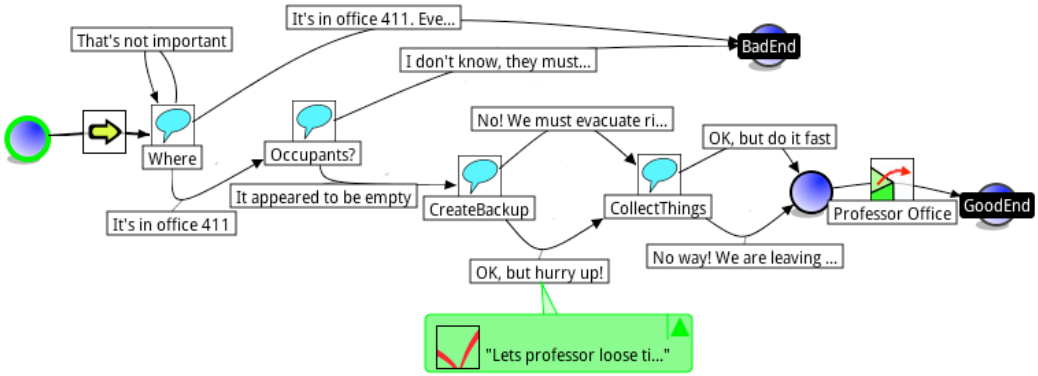


Figure 60 The "ConversationWithProfessor" story-part in the fire protocol game

Playing the game

The game, using the existing graphic resources, can be played directly from WEEV (Figure 61).

Figure 61 The fire protocol game during game-play

3. The Hematocrit game

The *hematocrit* game is a point-and-click simulation of a medical procedure. This game was developed with the School of Medicine at the Complutense University of Madrid. The procedure is taught in a compulsory course of medicine studies. This game was used to evaluate the learning outcomes of students, used in different case studies (Pablo Moreno-Ger, *et al.*, 2010; J. Torrente, Moreno-Ger, P., Fernández-Manjón, B. & del Blanco, A., 2009).

The game takes place in a laboratory, where the student must interact with different elements (e.g. test tubes) and machinery (e.g. centrifuge) to analyze the *hematocrit* level (HCT) of a blood sample. In-game evaluation allows the teacher to establish how the game was played and the student to rethink wrong assumptions and correct mistakes.

The basic procedure is as follows:

1. Put on a pair of clinical gloves
2. Shake the blood container gently to homogenize the sample and fill a capillary with blood
3. Seal the colored end of the capillary tube with plasticine
4. Place the sealed capillary tube into the centrifuge
5. Centrifuge the sealed capillary tube for five minutes
6. Measure the length of the packed cell volume and the total length of the sample in the capillary tube, obtaining the HCT value as a percentage

This game is considerably more complex than the other two games presented. The original <e-Adventure> representation has around 25 scenes, while the other two games had 2 and 9 respectably.

World

The world in the *hematocrit* game includes the different spaces where the player needs to perform the different parts of the procedure (Figure 62). Some spaces do not exactly fit the narrative metaphor, but are needed to simplify the representation (i.e. TiltedTestTube).

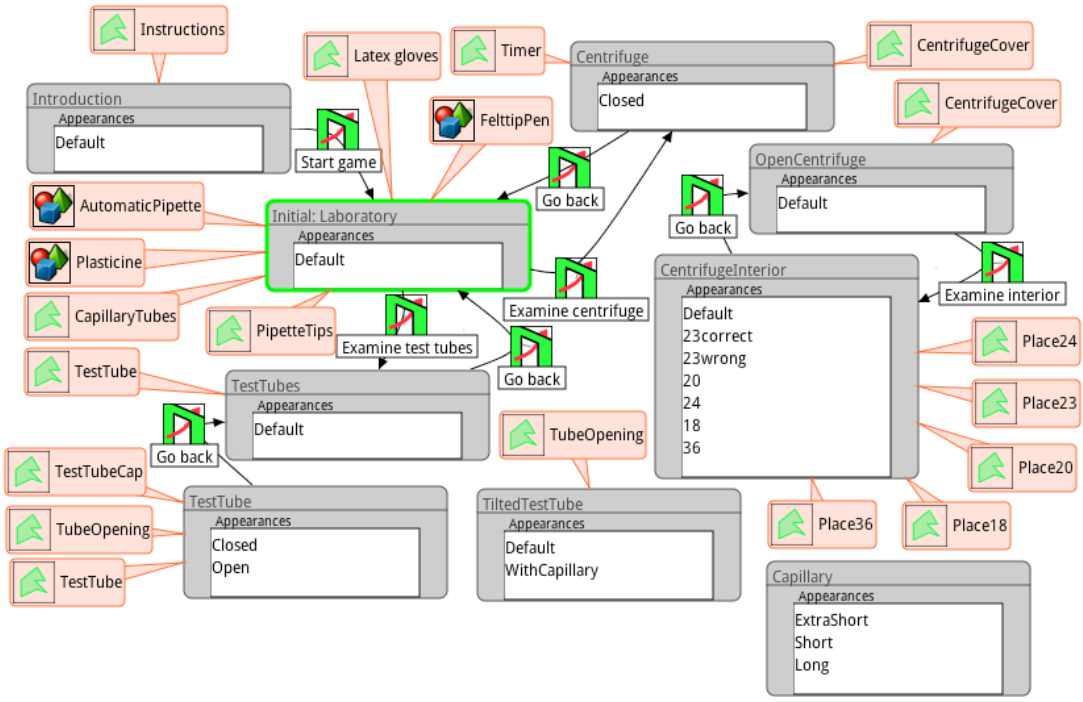


Figure 62 The world in the hematocrit game as represented in WEEV

Story

The basic story for the *hematocrit* game uses several story-parts to represent different steps in the game (Figure 63). The high-level representation presents both a path that leads to a good ending and several incorrect paths that force the player to start over.

The story-parts used in WEEV do not exactly fit the steps as described earlier, but in some cases represent 2 steps in the procedure:

- *PutOnLatexGloves*: step 1 in the procedure
- *FillCapillaryTube*: steps 2 and 3 in the procedure
- *SpinCapillaryTubeInCentrifuge*: steps 4 and 5 in the procedure
- *MeasureHCT* and *MeasureHCTAnemic*: two variations of step 6 in the procedure

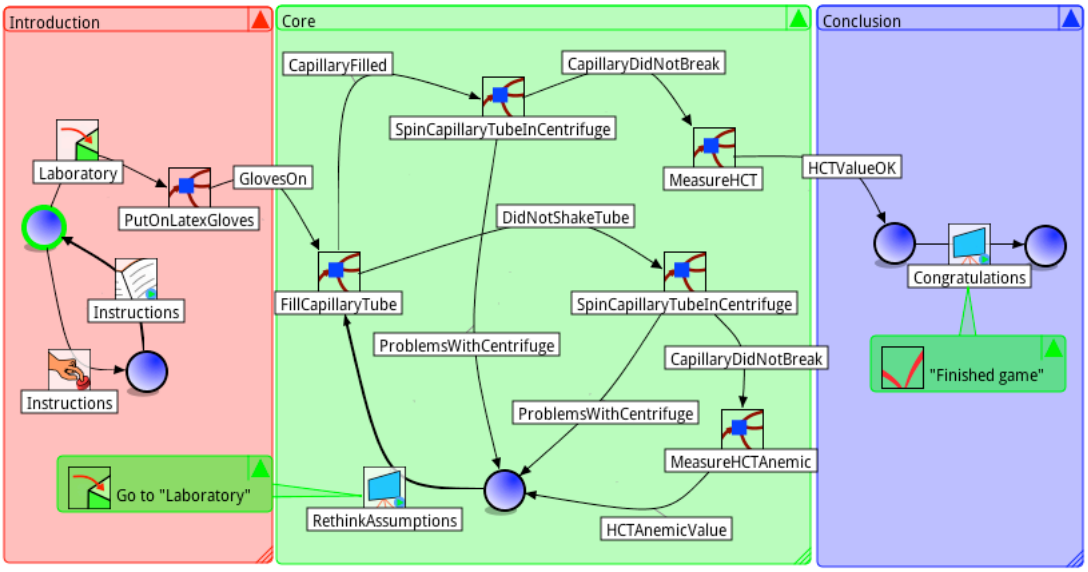


Figure 63 The main storyline in the hematocrit game

Story-Part: PutOnLatexGloves

As the first step in the procedure, this story-part forces the users to place on the latex gloves (Figure 64). If the user tries to do anything else, a message will be shown on the screen and a log line will be created in the assessment report.

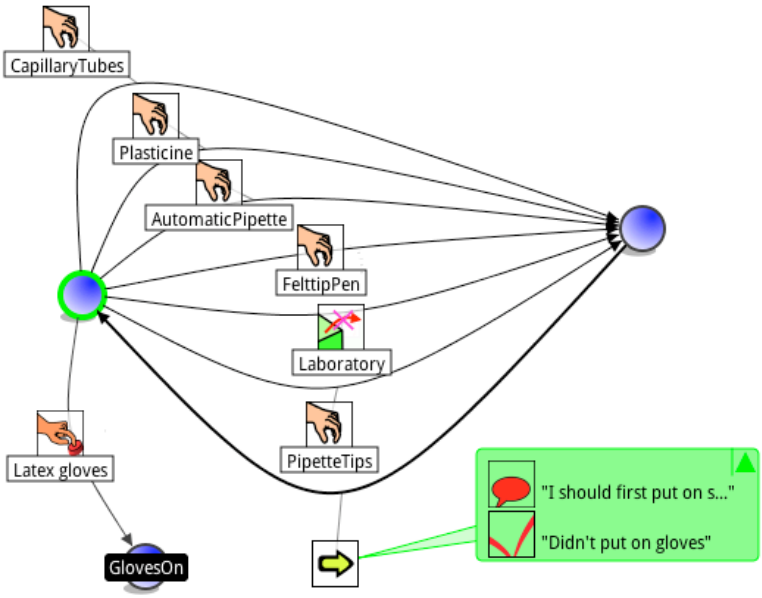


Figure 64 The "PutOnLatexGloves" story-part in the hematocrit game

Story-Part: FillCapillaryTube

This story part deals with all the steps in the procedure related to filling the capillary tube with blood (steps 2 and 3). The representation includes custom actions (e.g. “Shake”) and other story-parts to simplify it (Figure 65).

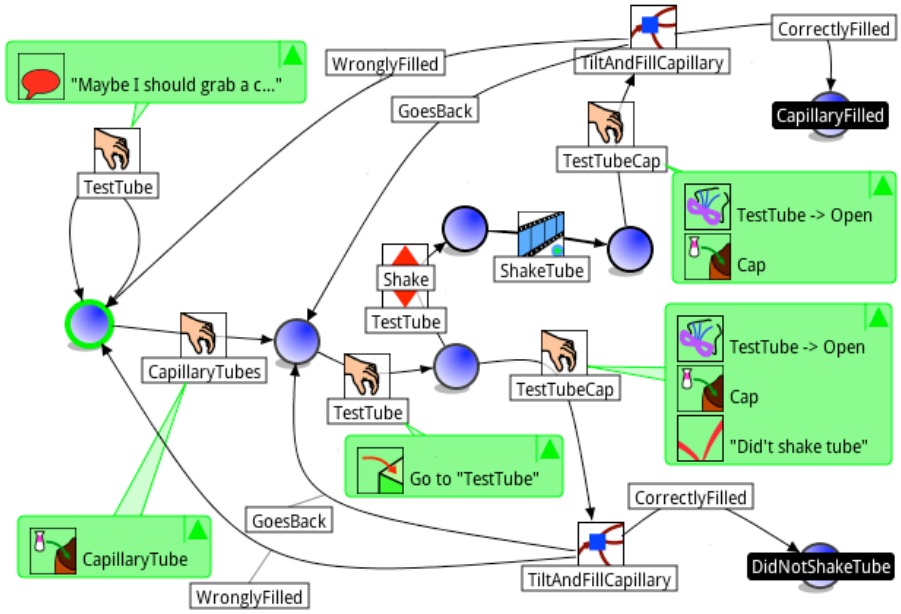


Figure 65 The "FillCapillaryTube" story-part in the hematocrit game

Story-Part: TiltAndFillCapillary

The “TiltAndFillCapillary” story part deals with some details related to filling the capillary tube. For example, the player might place the capillary tube inside the test tube before tilting it. The representation uses custom actions (i.e. “Tilt”) and multiple-choice questions (e.g. “CapillaryPlacement”) to add details to the game (Figure 66).

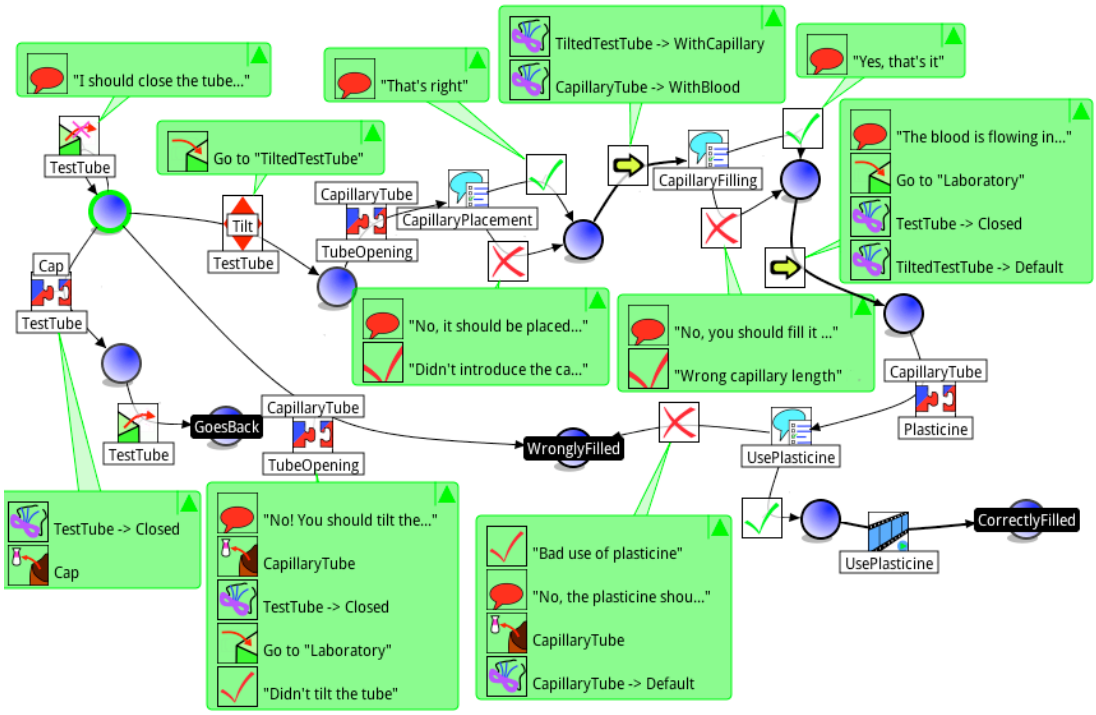


Figure 66 The "TiltAndFillCapillary" story-part in the hematocrit game

Story-Part: SpinCapillaryTubeInCentrifuge

This story-part deals with parts 4 and 5 of the procedure. The student must learn to place the capillary in the correct position, to set the centrifuge and to remember to recover the correct capillary from the centrifuge (Figure 67).

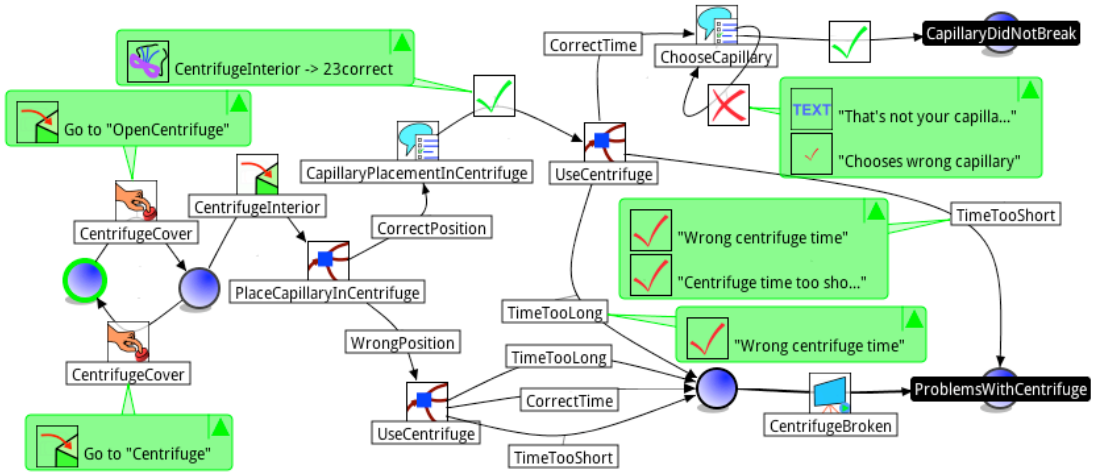


Figure 67 The "SpinCapillaryTubeInCentrifuge" story-part in the hematocrit game

Story-part: PlaceCapillaryInCentrifuge

This story-part allows a more detailed description of the placement of the capillary tube in the centrifuge. The player can choose to place the tube in the correct position (that depends on the placement of other tubes in the centrifuge) or can wrongly place the tube in a position that will unbalance the centrifuge (Figure 68).

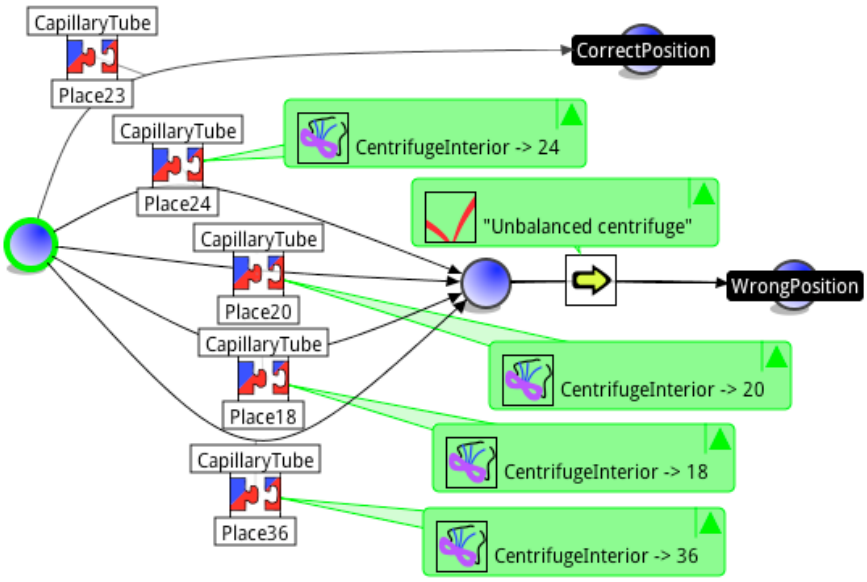


Figure 68 The "PlaceCapillaryInCentrifuge" story-part in the hematoric game

Story-part: UseCentrifuge

The centrifuge must be closed and the timer set for it to work. This step describes both the actions by the player and the feedback, in the form of a cut

scene, which indicates to the player that the centrifuge is working (Figure 69).

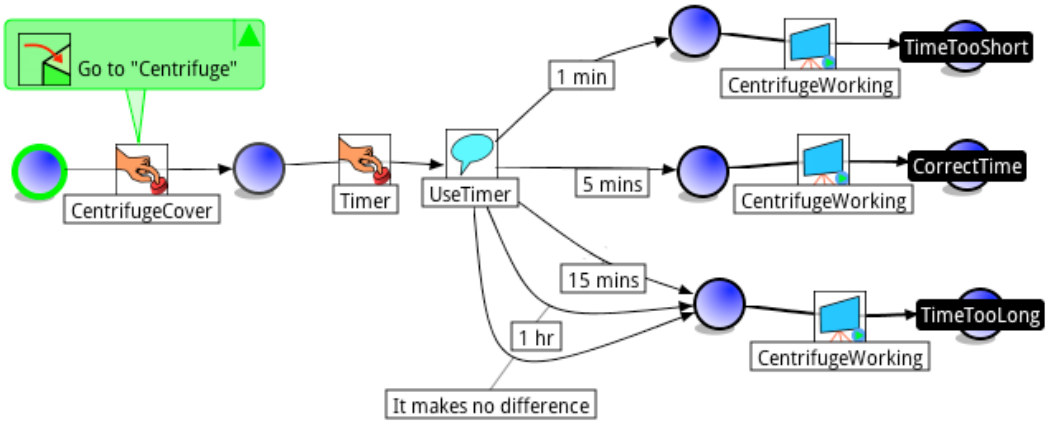


Figure 69 The "UseCentrifuge" story-part in the hematocrit game

Story-part: MeasureHCT

This story-part, only reached if the previous steps in the procedure where preformed correctly, asks the player to measure the value shown by two samples (Figure 70). This was a design decision, that doesn't exactly fit with the "story" this far but it was included to make sure the students preformed the measurement correctly. A random node is used to change the order in with the questions are asked.

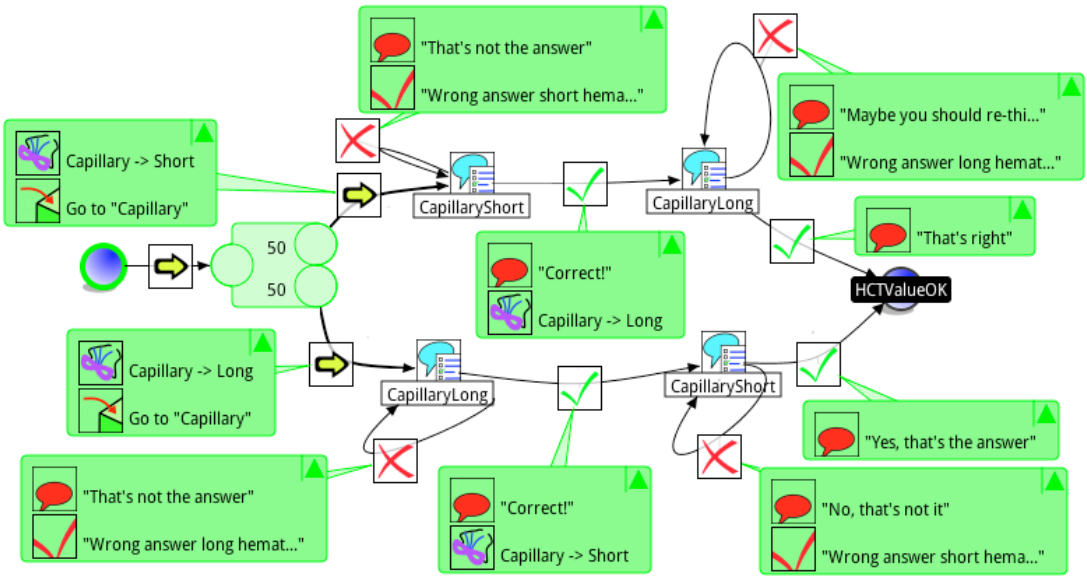


Figure 70 The "MeasureHCT" story-part in the hematoric game

Story-part: MeasureHCTAnemic

The player will only reach this story-part when performing all steps correctly but failing to shake the test tube before extracting the sample. The user is asked to measure the *hematocrit* level of a blood sample that shows the consequences of not shaking the tube: the *hematocrit* level will be abnormally low (Figure 71).

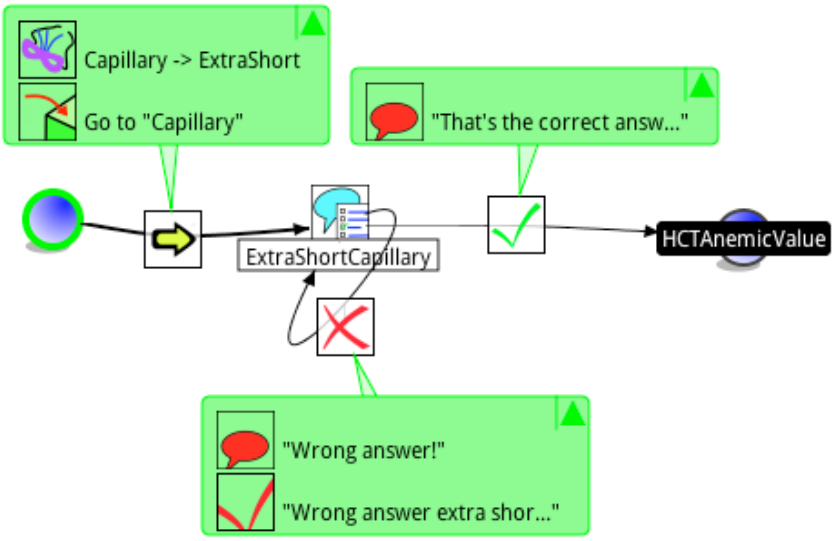


Figure 71 The "MeasureHCTAnemic" story-part in the hematoric game

Playing

The *hematocrit* game can be recreated using the original resources (Figure 72).

Figure 72 The hematoric game during game-play

4. Conclusions of this chapter

This use cases show the expressiveness of the WEEV system and WEEVL (the visual language) in particular. The games here represented show different degrees of complexity. The WEEV system allows the user to deal with such complexity by the use of story-parts, multiple-choice questions and other representation enhancements.

Some of the most complex representation can still be problematic for the beta version of the system during compilation if used simultaneously, although all parts work correctly independently. This shows the complexity of transforming a game described using a visual language into a content-centric specification such as the one used by <e-Adventure> where the logic is scattered in the different components.

The *hematocrit* game shows that the representation might still result in games that are too complex for inexperienced users to understand directly. However, it must be noted that the original's game logic uses hundreds of panels within panels, while the information in WEEV needs just 8 story-parts and the main storyline.

CHAPTER VIII. FORMATIVE EVALUATIONS

This chapter presents the two formative evaluations used to establish the perception of the system by real users, their ability to understand the representations of games and a measure of the usability of the system. The full results of the surveys in the evaluations are included in Appendix sections of this thesis.

1. First formative evaluation

Objectives

This evaluation, the first formative evaluation with users, was an attempt to establish how different users interpreted the representation of games presented by the WEEV system, how they used the system and how they perceived the potential of the system. At the same time, an effort was made to establish which were the most problematic parts of the system and the representation to guide future efforts into improving them.

Evaluation set-up

The first formative evaluation was performed by users with different degrees of knowledge of the <e-Adventure> platform or other video game development tools. This evaluation was performed over the Internet, allowing users to answer the questions and perform the different task using their own computers and time them as they pleased.

The process followed for the evaluation was:

- The users where asked questions about their general knowledge of video games and video game platforms.
- The users where presented with a representation of a game created by WEEV and asked to describe about what they could identified.
- Later, the users where given a tutorial, a guided experience and some ideas to create another simple game. They where asked to upload the resulting games.
- After using the system, the users where asked a series of subjective questions to establish their opinion of the system. This questions where about the system in itself and in comparison to <e-Adventure> (for those users familiar with that platform).

10 users participated in the first two phases of the evaluation. The last two phases where only completed by 9 of these users. Three groups of questions in the second survey dealt with the difficult of the system, the general perception of the WEEV system and a contrast between WEEV and the regular <e-Adventure> editor (Table 1).

Table 1 Wording of the questions in the second survey

Question		Wording
Q1	*Difficulty to use WEEV: 1-Very difficult and 5-Very easy*	
	Q1.1	The game creation process with <e-Adventure> WEEV is:
	Q1.2	Understanding the icons used in the buttons in WEEV is:
	Q1.3	Understanding the representation and metaphor used by WEEV is:
	Q1.4	Designing an attractive story for a game with WEEV is:

	Q1.5	Implementing the story of a game with WEEV is:
	Q1.6	Using WEEV for a novel developer would be:
	Q1.7	Understanding the tutorial is:
Q2	*General perception of the WEEV system: 1-Strongly agree and 5-Strongly disagree*	
	Q2.1	I believe WEEV is easy to use
	Q2.2	I believe that the metaphor and representation are clear to understand
	Q2.3	I believe that the representation allows a clear understanding of the story
	Q2.4	I believe that the icons used correctly fit what they represent
	Q2.5	I believe that a novel user would find WEEV easy to use
	Q2.6	I believe that the new interface (buttons, dialogs, etc.) is easy to use
	Q2.7	I believe the new interface (buttons, dialogs, etc.) is easily understood
	Q2.8	I believe the new interface (buttons, dialogs, etc.) is visually pleasant
	Q2.9	I like the name WEEV (Writing Environment for Educational Videogames)
Q3	*Comparison with <e-Adventure>: 1-Strongly agree and 5-Strongly disagree*	
	Q3.1	I believe that WEEV is easier to use than <e-Adventure>
	Q3.2	I believe that WEEV eases the creation of the world in games
	Q3.3	I believe that WEEV eases the creation of the story of the games
	Q3.4	I believe that WEEV can completely substitute <e-Adventure>
	Q3.5	I believe that integrating WEEV in the <e-Adventure> platform, creating a single distribution, would be positive because it will benefit game development

Results

The full results of the surveys from this evaluation are included in *Appendix A*. However, the most relevant results are included here.

- Intuitive understanding of the world representation: Using the textual description given by the users, their interpretations of different aspects were evaluated. Table 2 shows the percentage of users who correctly identified different elements.

Table 2 Intuitive understanding of different representation elements in the first evaluation

Concept	% of users
Understood that the screen shot represented the game world	90%
Could explain correctly that it represented scenes or spaces and that the links represents paths from one to the other	80%
Took notice that the world representation included the elements in each space and identified its meaning correctly	70%

- Intuitive understanding of the story representation: Using the textual description given by the users of the system, their interpretations of different aspects were evaluated. Table 3 shows the percentage of users who correctly identified different elements.

Table 3 Intuitive understanding of different story representation elements in the first evaluation

Concept	% of users
Understood that the screenshots represented the full description of the story-flow of a video game	60%
Took notice that the description belonged to an educational video game (and not just any video game)	20%
Correctly identified the states as game states	50%
Correctly identified the transitions as actions available to the user	90%
Correctly identified the use properties of the transitions as effects of the actions	60%
Correctly identified the encapsulation of content of the main story (first screenshot) as the contents of the second screenshot ("story part")	50%
Took notice of the use of narrative structure elements to organize the story	30%
Identified the use of timers in the story-flow and their meaning	10%
Identified and correctly explained all the representation enhancements used	0%

- Answers to the questions in the second survey: The second survey included questions regarding different aspects of the perception of users of the WEEV system (Table 1). The descriptive values to the answers to each of these questions and the overall values for each group of questions are included in Table 4.

Table 4 Descriptive values of the answers to the questions in the second survey of the first formative evaluation

Question	N	Mean	Std. Dev.	Min	Median	Max
Q1 (overall)	**9**	**22.80 / 35**	**3.07**	**18**	**24**	**26**
Q1.1	9	3.00	0.50	2	3	4
Q1.2	9	3.44	0.73	2	4	4
Q1.3	9	3.11	0.78	2	3	4
Q1.4	9	3.44	0.73	2	4	4
Q1.5	9	3.33	0.71	2	3	4
Q1.6	9	2.67	1.00	1	3	4
Q1.7	9	3.78	0.44	3	4	4
Q2 (overall)	**9**	**32.70 / 45**	**5.96**	**21**	**34**	**40**
Q2.1	9	3.56	0.73	2	4	4
Q2.2	9	3.44	0.88	2	4	4
Q2.3	9	3.89	0.93	2	4	5
Q2.4	9	3.78	0.97	2	4	5
Q2.5	9	3.00	1.12	1	3	4
Q2.6	9	3.67	1.32	2	4	5
Q2.7	9	3.44	1.13	2	3	5
Q2.8	9	3.56	1.33	1	4	5
Q2.9	9	4.33	0.50	4	4	5
Q3 (overall)	**8**	**18.60 / 25**	**2.88**	**14**	**19**	**22**
Q3.1	8	3.50	1.07	2	4	5
Q3.2	8	3.88	1.25	2	4	5
Q3.3	8	4.50	0.53	4	4.5	5
Q3.4	8	2.25	1.04	1	2	4
Q3.5	8	4.5	0.76	3	5	5

Discussion

The results of the survey before the use of the system, based solely on captures of the world and the story of a game represented using WEEV show that users mostly understood correctly the representation used for the world. The elements in the story show different results, that at first glance seem less promising given that even if most users clearly understood what it was representing (60%) no user could fully describe all the representation elements used. However, it must be noted that this was based just on captures of the representation, with no previous feedback about what the systems intention was (even if all users knew it had something to do with video games) and that they where not asked about particular elements but asked to freely write a description of what they saw.

The results of the second survey, completed after using the game, provide better information about how the users perceived the system as a whole after following a tutorial and a guided game creation experience. These results show that:

- Users found most of the system easy to use in general (22.80 / 30) even if they considered that a novel user would not find it particularly easy (2.67 / 5).
- Users provided a positive perception of the system in general, mostly agreeing with the provided statements (32.70 / 45). They found the name of the system particularly fitting (4.33 /5) and generally considered that the representation allowed a clear understanding of the story (3.89 / 5).
- Users who where familiar with <e-Adventure> (9 users) generally agree with the statements (favorable to WEEV) in the comparison (18.60 / 25) although they mostly considered that it could not completely replace the current <e-Adventure> editor (2.25 / 5) they strongly believed that it eased the creation of the stories and that it could be a positive complement to the platform (4.5 / 5 in both cases).

Conclusions

The results of the first formative evaluation showed that there was still wide room for improvements in the wizard, the representation, and the system as a whole. The users found some elements particularly confusing (e.g. the multi-interaction representation) and found it hard to use some of the aspects that should have been particularly easy in WEEV (e.g. the use by novel developers).

These results, however, have been helpful even if not particularly positive as they shown that even with the problems the users saw the potential of the system as a tool to complement the current <e-Adventure> approach and to ease the creation of stories, the main aim of the WEEV system. Besides, they provided an insight into what needed small improvements (e.g. the world edition), further work (e.g. the wizard) or a complete rethinking (e.g. the multi-interactions).

2. Second formative evaluation

Objectives

This second formative attempted to study the usability of the system, evaluate the improvements in the representation since the first evaluation and establish the perception of the system (particularly in comparison <e-Adventure>) of users with some experience as educational video game developers.

Evaluation set-up

The second formative evaluation was presented to students in an educational video game development seminar. As the evaluation was performed in the previous to last lecture of that seminar, these users can be considered "experts" in, or at least knowledgeable about, educational video game development. The procedure used for this evaluation, as it was performed in a controlled environment (i.e. a laboratory at the university) (Figure 73), used a more systematic approach in comparison to the one used for the first formative evaluation.

Figure 73 Users evaluating the WEEV system in a laboratory session

In this case the procedure used was the following:

- The users where given a brief introduction to the system, including the metaphor and the goals. They were not, however, taught how to use it or the representational elements in the system.
- The users where asked to create a game following a guided experience script. If the users had spare time, they were allowed to use the system in different ways, such as creating their own game or using the system to recreate one of the games created during the seminar. The aim was for the users to familiarize themselves with the tool, explore its possibilities and use the help system when in trouble. This lasted for around 50 minutes.
- After using the system, the users where asked to fill out two surveys. A usability survey based on the one proposed by Lewis (1995) and a subjective survey about their opinion of the system and a comparison with the <e-Adventure> platform with which they where familiar.
- The users were asked to upload the games they created, from which usage data (e.g. time in each panel) was extracted. This was possible because the version used in this formative evaluation was modified to

recollect and time-stamp all interactions of the user with the system. Users where informed of this fact before the evaluation.

20 users participated in this evaluation, of which 19 uploaded the game they had created for further statistical analysis. From the upload games, a profile of the times spent in different parts of the games was extracted and allowed to establish any bug the player found in the development. Besides, a free question in the surveys allowed users to explicitly state any bug they found.

The survey about the opinion of users tried to establish their perception of the different parts of the system. The questions in this survey, grouped by subject, are detailed in Table 5.

Table 5 Wording of the questions and identified underlying factors in the general perception questionnaire

	Question	Wording
P1	*Understanding of the metaphor and representation*	
	P1.1	The metaphor (i.e. games considered as stories) is clear to understand
	P1.2	The representation (i.e. how things are displayed on the screen) is clear to understand
	P1.3	The story representation is clear to understand
	P1.4	Icons are adequate for what they represent
P2	*Usefulness of the WEEV system*	
	P2.1	A novel user would find WEEV easy to use
	P2.2	An integration of WEEV in the <e-Adventure> platform would greatly easy game development
	P2.3	WEEV is useful to document a game
	P2.4	Adding educational features is very easy using WEEV
P3	*Comparison of WEEV with <e-Adventure>*	
	P3.1	WEEV is easier to use than <e-Adventure>
	P3.2	Creating stories with WEEV is clearer than with <e-Adventure>
	P3.3	WEEV can completely replace <e-Adventure>
	P3.4	Adding educational features with WEEV is easier than with <e-Adventure>

Results

The full results of the surveys from this evaluation are included in *Appendix B*.

Most users (19/20) had enough time to fully implement the proposed game but one user failed to upload the files needed to verify this. Using information embedded in the games, we can establish the users spent an average of 53.12 minutes (Std Dev 10.96) using the system. Of this time, 32.22 minutes (Std Dev 8.23) were spent in the story edition panel, representing just over 60% of the total time. The full descriptions of the times spent editing the actors, the world and the story, along with the percentage of the total time they represent (variables ACTORS%, WORLD% and STORY%), are included in Table 6.

Table 6 Descriptive values of the times spent editing the actors, the world and the story (in seconds and as a percentage of the total time)

	Mean	Std Dev	Min	Median	Max
TOTAL (s)	3205	583.44	2007	3328	4105
ACTORS (s)	374.16	157.85	102	330	696
WORLD (s)	715.95	289.92	120	663	1460
STORY (s)	1869	555.38	713	1936	2730
ACTORS%	11.86	5.95	5.08	10.24	31.10
WORLD%	22.16	8.53	5.98	21.50	49.54
STORY%	58.58	15.64	24.19	63.33	86.55

Table 7 shows the valuation of the users of the different aspects of the system usability. OVERALL represents the overall answers, SYSUSE represents the System Usefulness, INFOQUAL the Information Quality and INTERQUAL the Interface Quality, according to the factor analysis in made by Lewis (Lewis, 1995).

Table 7 Descriptive values of the underlying factors in the usability survey

	N	Items	Mean	Std Dev	Min	Median	Max
OVERALL	20	19	69.45	7.66	57	68.5	86
SYSUSE	20	8	32.25	3.48	27	33	39
INFOQUAL	20	7	21.70	4.01	14	22	32
INTERQUAL	20	4	15.50	2.42	11	15	20

As the evaluation was performed using an alpha version of the system, we performed a Wilcoxon test to establish if there was a significant effect in the valuation of the system for users that found bugs (7) against those who did not (12), finding no significant statistical difference (Pr 0.1552 > 0.05). Besides, as the time the users had with the system was fixed, we controlled for the effects of not finishing the all the steps in guided experience (due to either problems with the system or that the users found it harder than others) on the results using a Wilcoxon test and found no statistically significant effect (Pr 0.9636 > 0.05).

Table 8 shows the descriptive values of the factors in the perception survey. This factors where created based on the content of the questions because the sample size was too small to perform a full factor analysis (14 <= n <= 17).

Table 8 Descriptive values of the perception factors

	N	Mean	Std. Dev.	Min	Median	Max
P_TOTAL	**17**	**41.76 / 60**	**9.42**	**9**	**15**	**55**
P1	**17**	**16.29 / 20**	**1.86**	**13**	**16**	**19**
P1.1	17	4.12	0.93	2	4	5
P1.2	17	4.12	0.60	3	4	5
P1.3	17	4.18	0.64	3	4	5
P1.4	17	3.88	0.79	3	4	5
P2	**17**	**13.29 / 20**	**3.50**	**5**	**14**	**17**
P2.1	17	3.71	0.99	2	4	5
P2.2	14	4.14	1.17	2	3	4
P2.3	14	3.21	0.70	2	3	4
P2.4	15	4.00	0.66	2	3	4
P3	**14**	**14.79 / 20**	**2.72**	**9**	**15**	**19**
P3.1	14	4.14	0.77	3	4	5
P3.2	14	4.14	0.77	3	4	5
P3.3	14	3.00	1.18	1	3	5
P3.4	14	3.77	0.73	3	4	5

Discussion

From technical standpoint, the test version of the WEEV systems used during this evaluation performed correctly. The bugs reported during the formative evaluation did not stop the users from being able to take full advantage of the implemented functionalities. The results show that the problems with the

system (i.e. finding a bug) had no significant influence over the results. Users were able to test all the functionalities including the creation of playable educational video games from the descriptions created with WEEV.

The usability survey results are satisfactory, most users found the system usable overall (69.45 / 95), finding the lower valuation in the Information Quality factor (21.70 / 35). The informal conversations with the participants showed this, as many users complained that the system did not allow certain things (e.g. placing effects in a story state) without indicating what was happening. We expect that the results of this survey can be further improved by including the comments made by the users regarding their experience, concentrating in the information provided to the users during the interaction with the system.

The general perception survey shows that users understood correctly the representation and metaphors used (16.29 / 20). The users did not find the usefulness of the system as high as could be expected (13.29 / 20), however it must be noted that this value has a high standard deviation (3.50) indicating that not all users agreed upon this fact. We considered this might be due to the fact that the users where programmers and did not find the explicit representation and guided approach introduced by WEEV useful given their high technical skills. In this regard, the users did not find WEEV to be particularly useful to document games (3.21 / 5) even if they did find the representation helpful and easy to understand. We believe that the wording of the question might have been confusing and that users might have understood that they were being asked about using WEEV to document previously existing games and not as an additional feature of games developed directly using WEEV.

The users that participated in the survey and had used the <e-Adventure> platform during the seminar found that WEEV was better over that platform at least for the tasks it was designed to do (14.79 / 20). This result is significant because WEEV is designed specifically as an improvement over <e-Adventure> in the regards considered in the questions of this group. It is particularly relevant that users found that WEEV was easier to use both overall and to describe the stories of games (4.14 / 5 in both cases). However, the fact that adding educational features was found slightly less easy (3.77 / 5) might be attributed to a problem in the code that caused the assessment rules added to the games not to work correctly during the evaluation.

Conclusions

The results of this second formative evaluation appear to be much more satisfactory than ones of the first formative evaluation, however no statically comparison can be performed. This difference mainly reflects the improvements made as a consequence of the feedback of the first evaluation.

It is of particular interest to note that the users correctly understood the metaphor and representation used, and that they considered the system useful for the task it was created to perform better than other systems (i.e. create stories).

The study of the time spent by users in the different panels shows how future development efforts should be focused. As expected, users spent most of their

time editing the story. However, a significant amount of time (>20%) was spent creating the world, even if the one needed for the guided experience was very simple. Besides, this variable also shows an important standard deviation, indicating that some users were able to perform this more efficiently. This might indicate that the world edition has to be improved, in order to reduce the time spent creating the world and allow users to focus more on the story.

3. Consequence of the evaluations

The iterative implementation method, the formative evaluations and the overall experimental approach to development resulted in different instances of the system being created, tested and improved upon. Some of these improvements are relevant because they show different approaches to the same problem or wrong approaches that could be spotted thanks to this development model. The evaluations allowed the correct identification of problems in the wizard and the WEEVL language, producing an "evolution" in the system.

Evolution of the wizard

The basic steps in the wizard where clear from the onset, being inspired in theory or resulting from technical necessity (e.g. some settings could not be changed once the user started creating the story without mayor problems to be accounted for). However, each step in the wizard presented some unique problems (e.g. how to better describe the games, by their potential use or the kind of interactions?) and general ones (e.g. how much information must be provided to the user?).

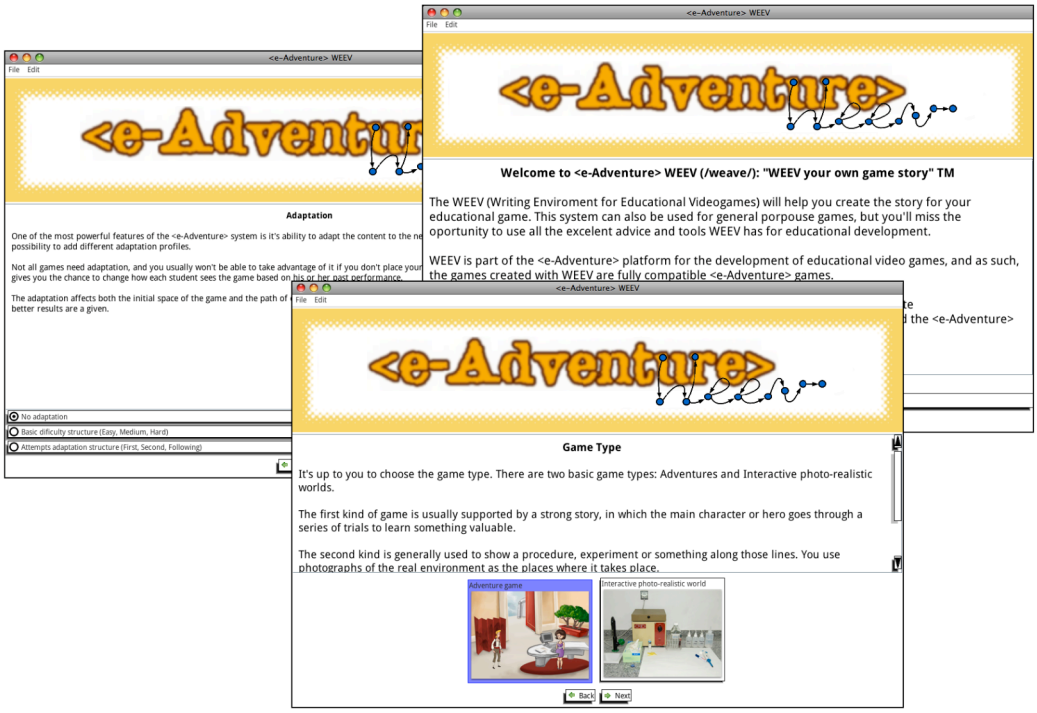


Figure 74 Some panels of the WEEV wizard in an old version. Long explanations and no contextual help button can be appriciated

The most relevant evolution in the wizard, from the first versions to the current interaction was in how the information is displayed to the user. Early versions of the system (Figure 74) provided detailed descriptions of each step and the meaning of every option available to the user. This proved to be problematic as most users just rushed through the screen without reading the information.

Current versions of the wizard (Figure 75) provide just the minimal information in the screen relevant to completing each step. However, the user can access a full description of the step, with detailed information on every choice though contextual help available using a help button always visible on the screen.

Figure 75 Some panels of one of the latest verions of the WEEV wizard. Short and clear explanations, along with a contextual help button are visible

Evolution of the language

The current WEEVL representation elements have been developed and improved across several iterations. The basic language was created based on proposals of narrative and video game studies. However, such proposals usually approach the problem from a theoretical perspective, implying different complications in the actual representation.

An example of the evolution suffered by the language can be found in the changes to what is called the “Multi-interaction” element in WEEVL (Figure 76). One of the first representation attempts for this element was so complex that it included more than 5 different configuration parameters for each branching that it allowed, creating a virtually exponential increase in complexity when something was modified. During the formative evaluation it caused so many problems to the users that not one could use it or understand its use in other games correctly.

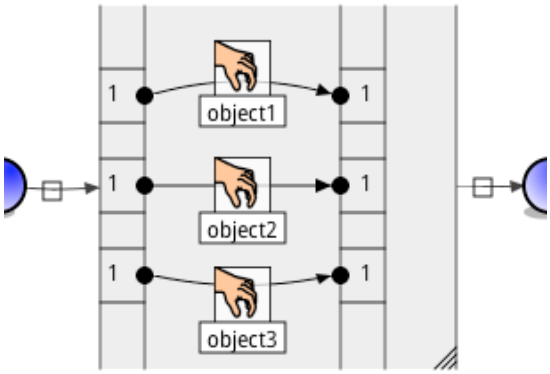


Figure 76 Deprecated representation of "multi-interactions". This representation used different parameters that left the underlying state machine metaphor making it very difficutlt to understad and use.

The new representation, used in the second formative evaluation, proved to be much better (this representation is the one described in Chapter V). The users found the representation intuitive and easy to understand, proving that the evolution based on formative evaluations can help improve the perception of the system by users.

4. Conclusions of this chapter

The formative evaluations show that users that used the tool find it potentially useful. Besides, these formative evaluations were helpful to discover where improvements were needed the most as well as many bugs (which have been addressed since then). The progress shown in the results from one evaluation to the other indicated that the system was greatly improved thanks to the feedback from users and makes apparent the benefits of an iterative development process for the system.

CHAPTER IX. CONCLUSIONS AND FUTURE WORK

This chapter presents a discussion of this work, as well as the contributions of this thesis and the work expected to be undertaken regarding the WEEV methodology and system in the future.

1. Summary

This thesis presents the WEEV (Writing Environment for Educational Video games) system, within a detailed theoretical framework and contextualized with other systems that share one or more features with it. The system, based on a narrative metaphor, uses a step-by-step methodology implemented as a wizard. The complex behaviors of games (e.g. user interaction) are described using DSVLs, which by using a graphic representation are easier to learn, understand and remember, in an effort to reduce the complexity of the system as a whole.

2. Discussion

The theoretical framework presented in this thesis is used as a base for the WEEV methodology, which allows for the creation of educational games through the use of a wizard and two DSVL. The WEEV methodology uses a narrative metaphor and narrative theory concepts, to construct games with emphasis on their story. The results of using this methodology were shown by the creation of three use cases, two of which are re-implementations of pre-existing <e-Adventure> games freely available on-line. The use cases show that, even if some structures need further simplification, the system can be used to create fully playable games that have interest in different fields. At the same time, the uses cases show that the system in its current state works best with procedural games and needs further improvements to be truly applicable to narrative or story-based games.

A revision of the literature and of the currently available tools for game development and content creation shows that no other system that implements such a metaphor is available. The WEEV system however is currently available and can be downloaded at the <e-Adventure> *sourceforge* website[15] and can be independently tested by users. Proposals similar in nature do exist, but usually take advantage of logical and representational elements of traditional game development tools already familiar to many users. In WEEV, however, the narrative metaphor is taken as far as possible completely overturning the <e-Adventure> development process, in an effort to help users to achieve the best possible results through its use. Unfortunately this imposes some restrictions on games, making some game constructions overly complicated. This problem is at least partially addressed by the use of "representation enhancements" in the story DSVL but remains an open issue in some other cases as the last use case shown in this work proves.

The formative evaluations of WEEV with actual users showed two main traits: users find that the tool is usable and that it has potential to be used in real scenarios; and progress between the first and second evaluations reflect that the rapid prototyping iterative development model chosen for the development of the system is showing promising results. The different design decisions are based both on user feedback and the theoretical framework for

---

[15] *http://sourceforge.net/projects/e-adventure/files/WEEV/WEEV_beta0.1.zip (retrieved June 21st 2010)*

this work, improving the results users can achieve using WEEV while keeping the underlying metaphor and concepts grounded on sound theoretical work. The formative evaluations also show that there is still much space for improvement.

The inclusion of different educational features within the same metaphor used to develop the story is a new approach in itself. The current implementation shows the viability of such an approach and users of the system had no problems in adding in-game assessment to the games they created (this was identified as an open issue in <e-Adventure>). This shows the potential of the approach for the specific task it was design: the development of educational video games. Unfortunately, these features are integrated into the system in such a way that might discourage casual use of the tool for creating purely entertaining games.

3. Contributions

The main contributions of this work are:

- *A theoretical framework was established* to support the use of narrative metaphors in the game development process, taking into account contributions from different fields and disciplines.
- *A systematic approach for educational video game development is presented*, based on narrative concepts, grounded on the theoretical framework, and inspired by research conducted in the video game, educational and narrative theory fields.
- The *concrete implementation of the approach to educational video game development* introduced in this work is presented. This implementation is the WEEV system currently available for download as an alpha version
- *A DSVL capable of expressing the inherit complexity of video games in as simple and straightforward a way as possible was defined*. This DSVL was validated in user evaluations and modified according to user feedback
- *The system has been tested by real users through the use of formative evaluations*, which show that it possesses a high degree of usability and is perceived as useful and with potential as a support tool in game development. These evaluations also allowed the identification, and posterior correction, of problems in the model, the representation of the story and the platform as a whole.

4. Future work

The platform is currently under development, as is the underlying model. Thus, future work is needed, and this includes the definition and creation of a "recommendation system" that would help the creator of the game to achieve better results. This can improve both the narrative aspects of the game (e.g. helping make a better use of narrative structures) and the educational aspects (e.g. improving the game according to evaluation or engagement criteria).

In an effort to further increase the usability of the system, a "Stencils"-like tutorial, which would helps a novel user create a simple game within the WEEV interface, is currently under development. Besides, other systems to help the user are also under works, such as a debug mode that allows the user to view the flow of the game while it is being played. Other technical aspects need further improvement, such as support for copy-paste and other usability elements. Optimizations to the whole system are being made to provide a more streamlined experience for the user.

We expect to be able to perform more evaluations of the system with actual users. The open beta version process just started in expected to help in this regard. Further evaluation is expected to help improve the system and to validate the metaphor for specific uses. Besides, more thorough analysis of all the information collected in the surveys and from the system itself will be conducted in the hope of extracting other relevant information.

WEEV is also being used internally, to validate the approach in different domains. A new medicine game is being developed using WEEV to teach basic first aid concepts to teenagers, and this will require the introduction of new elements in the representation that we hope will help develop other games in this field. The system is also being applied to the language teach domain, through the creation of situational English games. The actual use of the system is expected to provide advances in the representation that will improve the system.

REFERENCES

Academic ADL Co-Lab. (2004). Outbreak Quest: A 90-day Game Development initiative. Retrieved 21th June, 2010, from http://www.academiccolab.org/resources/documents/OutbreakQuest.pdf

Adams, P. C. (1998). Teaching and Learning with SimCity 2000. *Journal of Geography, 97*(2), 47-55.

Amory, A. (2006). Game object model version II: a theoretical framework for educational game development. *Educational Technology Research and Development, 55*(1), 51-77.

Amory, A., Naicker, K., Vincent, J., & Adams, C. (1999). The Use of Computer Games as an Educational Tool: Identification of Appropriate Game Types and Game Elements. *British Journal of Educational Technology, 30*(4), 311-321.

Boshernitsan, M., & Downes, M. (2004). *Visual Programming Languages: A Survey* (No. UCB/CSD-04-1368). Berkley, California: University of California.

Burgos, D., Moreno-Ger, P., Sierra, J. L., Manjón, B. F., Specht, M., & Koper, R. (2008). Building Adaptive Game-Based Learning Resources: The Marriage of IMS Learning Design and <e-Adventure>. *Simulation & Gaming, 39*, 414-431.

Cordova, D. I., & Lepper, M. R. (1996). Intrinsic Motivation and the Process of Learning: Beneficial Effects of Contextualization, Personalization, and Choice. *Journal of Educational Psychology, 88*(4), 715-730.

De Freitas, S., & Oliver, M. (2006). How can exploratory learning with games and simulations within the curriculum be most effectively evaluated? *Computers & Education, 46*(3), 249-264.

Del Blanco, A., Torrente, J., Moreno-Ger, P., & Fernández-Manjón, B. (2009). *A General Architecture for the Integration of Educational Videogames in Standards-compliant Virtual Learning Environments.* Paper presented at the 9th IEEE International Conference on Advanced Learning Technologies (ICALT 2009).

Dickey, M. D. (2005). Engaging by design: How engagement strategies in popular computer and video games can inform instructional design. *Educational Technology Research and Development, 53*(2).

Dickey, M. D. (2006). Game Design Narrative for Learning: Appropriating Adventure Game Design Narrative Devices and Techniques for the Design of Interactive Learning Environments. *Educational Technology Research and Development, 54*(3), 245-263.

Ferre, X., Juristo, N., Windl, H., & Constantine, L. (2001). Usability basics for software developers. *IEEE Software, 18*(1), 22-29.

Folmer, E., & Bosch, J. (2004). Architecting for usability: a survey. *Journal of System and Software, 70*(1-2), 61-78.

Frye, B., & Frager, A. M. (1996). Civilization, Colonization, SimCity: Simulations for the Social Studies Classroom. *Learning & Leading with Technology, 24*(2), 21-23.

Gee, J. P. (2003). *What video games have to teach us about learning and literacy*. New York ; Basingstoke: Palgrave Macmillan.

Göbel, S., Salvatore, L., Konrad, R. A., & Mehm, F. (2008a). *StoryTec: A Digital Storytelling Platform for the Authoring and Experiencing of Interactive and Non-linear Stories*. Paper presented at the ICIDS 2008.

Göbel, S., Salvatore, L., Konrad, R. A., & Mehm, F. (2008b). StoryTec: A Digital Storytelling Platform for the Authoring and Experiencing of Interactive and Non-linear Stories *Interactive Storytelling* (Vol. 5334). Berlin / Heidelberg: Springer.

Hunicke, R., & Chapman, V. (2004). *AI for dynamic difficulty adjustment in games*. Paper presented at the Proceedings of the Challenges in Game AI Workshop, Nineteenth National Conference on Artificial Intelligence (AAAI '04), San Jose, California.

Juul, J. (2005). *Half-Real: Video Games between Real Rules and Fictional Worlds*: MIT Press.

Kelleher, C., & Pausch, R. (2007). Using storytelling to motivate programming. *Communications of the ACM, 50*(7), 58-64.

Kiili, K. (2005). Digital Game-Based Learning: Towards an Experiential Gaming Model. *Internet and Higher Education, 8*(1), 13-24.

Lewis, J. R. (1995). IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use. *International Journal of Human-Computer Interaction, 7*(1), 57-78.

Lindley, C. A. (2005). Story and Narrative Structures in Computer Games. In B. Bushoff (Ed.), *Developing interactive narrative content: Sagas/Sagasnet reader*. Munich: High Text.

Malone, T. (1981). Toward a Theory of Intrinsically Motivating Instruction. *Cognitive Science, 5*, 333-369.

Moreno-Ger, P. (2007). *A Documental Approach for the Creation and Integration of Digital Games in Virtual Learning Environments*. Universidad Complutense de Madrid, Madrid.

Moreno-Ger, P., Burgos, D., Sierra, J. L., & Fernández-Manjón, B. (2008). Educational Game Design for Online Education. *Computers in Human Behavior, 24*(6), 2530-2540.

Moreno-Ger, P., Burgos, D., & Torrente, J. (2009). Digital games in e-Learning environments: current uses and emerging trends. *Simulation & Gaming, 40*(5), 669-687.

Moreno-Ger, P., Burgos, P., Sierra, J. L., & Fernández-Manjón, B. (2007). A Game-based Adaptive Unit of Learning with IMS Learning Design and <e-Adventure> *Lecture Notes in Computer Science, 4753*, 247 - 261.

Moreno-Ger, P., Fernández, R. F., Sierra, J. L., & Fernández-Manjón, B. (2009). Model-checking for Adventure Videogames. *Information and Software Technology, 51*(3), 564-580.

Moreno-Ger, P., Martinez-Ortiz, I., & Fernández-Manjón, B. (2005, 14-16 December 2005). *The <e-Game> project: Facilitating the Development of Educational Adventure Games*. Paper presented at the Cognition and Exploratory Learning in the Digital age (CELDA 2005), Porto, Portugal.

Moreno-Ger, P., Martínez-Ortiz, I., Sierra, J. L., & Fernández-Manjón, B. (2008). A Content-Centric Development Process Model. *IEEE Computer, 41*(3), 24-30.

Moreno-Ger, P., Sierra, J. L., Martínez-Ortiz, I., & Fernández-Manjón, B. (2007). A Documental Approach to Adventure Game Development. *Science of Computer Programming, 67*(1), 3–31.

Moreno-Ger, P., Torrente, J., Bustamante, J., Fernández-Galaz, C., Manjón, B. F., & Comas-Rengifo, M. D. (2010). Application of a low-cost web-based simulation to improve students' practical skills in medical education. *Int. J. Med. Inform., 79*, 459-467.

Resnick, M. (2008). Sowing the Seeds for a More Creative Society. *Learning & Leading with Technology, 35*(4), 18-22.

Robertson, J., & Good, J. (2004). *Children's narrative development through computer game authoring*. Paper presented at the IDC 2004.

Robison, A. J. (2008). The Design is the Game: Writing Games, Teaching Writing. *Computers and Composition, 35*, 359-370.

Rollings, A., & Adams, E. (2003). *Game design*. Indianapolis, IN: New Riders.

Ryan, M. L. (2006). Avatars of story (pp. 101-109): University of Minnesota Press.

Squire, K. D. (2005). Toward a theory of games literacy. *Telemedium, 52*(1-2), 9-15.

Torrente, J., Moreno-Ger, P., Fernández-Manjón, B., & Sierra, J. L. (2008). *Instructor-oriented Authoring Tools for Educational Videogames*. Paper presented at the 8th International Conference on Advanced Learning Technologies (ICALT 2008), Santander, Spain.

Torrente, J., Moreno-Ger, P., Fernández-Manjón, B. & del Blanco, A. (2009). *Game-like Simulations for Online Adaptive Learning: A Case Study*. Paper presented at the Edutainment 2009: The 4th International Conference on E-Learning and Games.

Van Deursen, A., Klint, P., & Visser, J. (2000). Domain-Specific Languages: An Annotated Bibliography. *ACM SIGPLAN Notices, 35*(6), 26-36.

Van Eck, R. (2006). Digital game-based learning: It's not just the digital natives who are restless. *EDUCAUSE Review, 41*(2), 16-30.

Vogler, C. (1998). *The writer's journey: Mythic structures for writers*. Studio City, CA: Michael Wiese Productions.

Wexler, S., Corti, K., Derryberry, A., Quinn, C., & Barneveld, A. V. (2008). *The eLearning Guild: 360 Report on Immersive Learning Simulations*.

Wolff, A., Mulholland, P., Zdrahal, Z., & Joiner, R. (2007). Re-using digital narrative content in interactive games. *International Journal of Human-Computer Studies, 65*(3), 244-272.

APPENDIX A: FULL FIRST FORMATIVE EVALUATION SURVEY RESULTS

The results of the first formative evaluation are here included for completeness only. As the survey was made among Spanish users, it is here presented in its original Spanish. However, translations are provided for the questions and some multiple-choice answers.

1. Pre-experience survey

This survey was presented to the users before they used the platform. It aimed to establish the knowledge of the users about the platform and games in general. It provided insight into the intuitive interpretation of the visual language.

1. ¿Cuanto tiempo dedicas por semana a jugar con video juegos de media? (*How much time do you spend playing computer games each week?*)

Menos de 1 hora (*Less than 1 hour*)	30%	3
Entre 1 y 5 horas (*Between 1 and 5 hours*)	**60%**	**6**
Más de 5 horas (*More than 5 hours*)	10%	1

2. ¿Tienes (o has tenido) alguna consola de video juegos en casa? Marca las que correspondan. (*Do you have (or had) any game console at home? Check the appropriate ones*)

PS3	10%	1
Xbox360	10%	1
Nintendo Wii	20%	2
PSP	20%	2
NintendoDS	**40%**	**4**
PS2	**40%**	**4**
Xbox	10%	1
Gamecube	10%	1
Otra	40%	4

3. ¿Conoces el género de las aventuras gráficas? (Por ejemplo, Monkey Island, Myst, Dracula, etc.) (*Are you familiar with the adventure game genre?)*

Si (*Yes*)	**100%**
No (*No*)	0%

4. ¿Que aventuras gráficas has jugado? (*Which adventure games have you placed?*)

Monkey Island (1, 2, 3, …)	**90%**	**9**

Myst	10%	1
Indiana Jones	60%	6
Sam and Max	60%	6
GrimFandango	40%	4
Otras…	60%	6

5. Ordena, según su importancia para ti, las siguientes características que conforman un video juego (1-Más importante, 5-Menos importante) (*Sort by the importance you give to them, the following features of a video game, with 1 being Most important and 5 Least important*)

El reto que impone (*The challenge it poses*)	**1.6**
El aspecto gráfico (*The graphics*)	3.0
Gráficos en 3D (*3D graphics*)	4.6
La historia (*Story*)	2.1
Inteligencia artificial (*Artificial intelligence*)	3.7

6. Conocimientos sobre video juegos (1-Nada, 5-Mucho) (*General knowledge about video games, with 1 being little and 5 being much*)

Conozco la estructura interna básica de un motor de videojuegos (distintos módulos que lo componen) (*I'm familiar with the interal structure of a game engine*)	3.3
Conozco cómo funciona una tubería gráfica 3D para juegos (*I'm familiar with the 3D drawing pipeline of video games*)	2.9
Conozco alguna tecnología directamente relacionada con el desarrollo de video juegos (OpenGL, DirectX, motores y librerías para desarrollo de juegos, etc.) (*I'm familiar with one or more technologies directly related to video game development*)	3.3
Conozco la estructura básica de un juego del tipo "aventura gráfica" (*I'm familiar with the basic structure of adventure games*)	**3.6**
Conozco alguna herramienta para crear aventuras gráficas (*I'm famliar with a tool designed for adventure game development*)	3.2

7. Sobre el conocimiento teórico de la herramienta <e-Adventure> (About your theoretical knowledge of the <e-Adventure> platform)

He oído hablar de <e-Adventure>, pero solo he visto esporádicamente algunas presentaciones, los tutoriales o la ayuda (*I heard about <e-Adventure>, but only browsed though presentations, tutorials o the contextual help*)	**70%**	**7**
Conozco ampliamente <e-Adventure> desde un punto teórico: filosofía de diseño (basado en contenidos), fundamentos científicos subyacentes, modelo de desarrollo, etc. (*I'm very familiar with <e-Avdenture> from a theoretical perspective*)	30%	3

8. Sobre el conocimiento práctico de la herramienta <e-Adventure> (*About your practical knowledge about the <e-Adventure> platform*)

Nunca he utilizado <e-Adventure> (*I never used <e-Adventure>*)	10%	1
He probado a abrir y modificar ligeramente juegos ya existentes con <e-Adventure> (*I tried to open and lightly modify existing games*)	**60%**	**6**
He creado (de forma individual o colaborativa) uno o más juegos con la herramienta (*I developed, alone or in a group, one or more games with <e-Adventure>*)	30%	3

9. ¿Has oído hablar de <e-Adventure> Weev (es una herramienta distinta a <e-Adventure>)? (*Have you heard about <e-Adventure> WEEV?*)

No tengo idea de que es (*I have know idea*)	**60%**	**6**
Si, tengo una idea básica (*Yes, I have a basic idea*)	30%	3
Si, conozco sus principios de funcionamiento, representaciones, etc. (*Yes, I'm familiar with it's principles*)	10%	1

10. ¿Has desarrollado (o participado activamente en el desarrollo) de algún video juego? (*Have you developed (or been actively involved) in the development of a video game?*)

Ningún video juego (*No video games*)	**40%**	**4**
Uno o dos video juegos (*One or two video games*)	30%	3
Más de dos video juegos (*More than two video games*)	30%	3

11. Según tus conocimientos de la plataforma (1-Muy difícil, 5-Muy fácil) (*From your knowledge of the platform (being 1-Very difficult and 5-Very easy*)

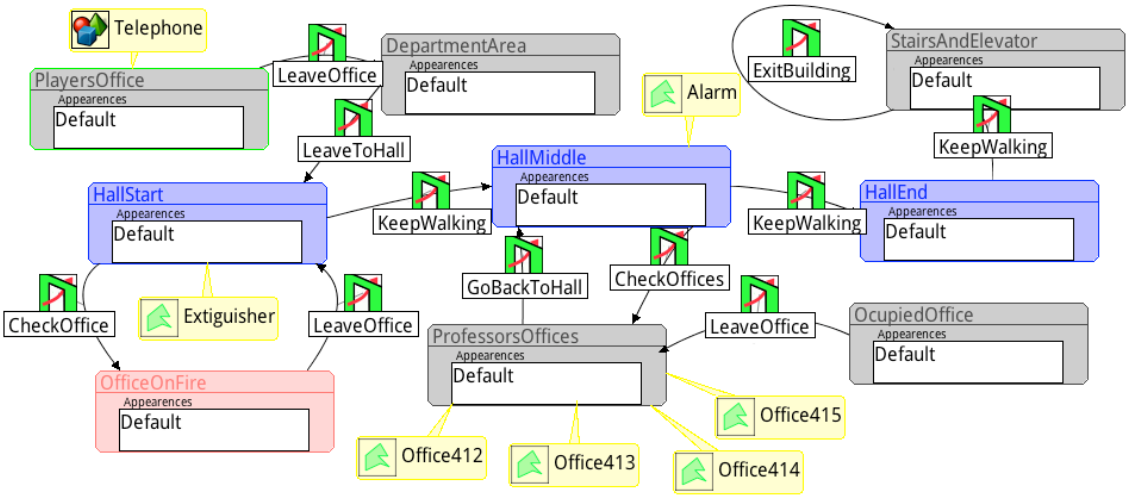
Cómo de fácil (o difícil) te parece <e-Adventure> (*How easy (or hard) do you think using <e-Adventure> is*)	**3.9**
Cómo de fácil (o difícil) te parece <e-Adventure> para una persona con un perfil poco técnico(por ejemplo, un profesor de secundaria, alguien sin conocimientos de programación, etc.) (*How easy (or hard) do you think a user with limited technical knowledge would find <e-Adventure>*)	2.7

12. Según tu experiencia, cuales te parecen los mayores problemas de <e-Adventure> a la hora de CREAR un video juego (sobre todo, para un usuario sin mucho conocimiento técnico - por ejemplo un profesor de secundaria) (*What do you considered the main problems with game DEVELOPMENT using <e-Adventure>*)

La obtención de recursos artísticos (fondos, animaciones de personajes, música, etc.) (*The creation of graphic resources*)	**80%**	**8**
Diseñar e implementar el "mundo" del juego (escenas y sus conexiones a través de salidas, zonas activas, etc.) (*Designing and creating the game world*)	20%	2
Diseñar e implementar un hilo narrativo complejo y atractivo (*Designing and creating a complex and attractive narrative*)	**60%**	**6**

El uso de condiciones (lógica, etc.) (*The use of conditions*)	50%	5
El sistema de efectos (por ej., lanzar una escena, reproducir un sonido, consumir objeto, etc.) (*The effect system*)	30%	3
Las acciones (e.g. coger, usar, hablar, etc.) (*The use of actions*)	10%	1
Diseñar e implementar conversaciones complejas (*Design and create complex conversations*)	40%	4

13. El siguiente grafo describe un aspecto concreto de un juego. Describe brevemente qué es lo que, según tu criterio, representa el gráfico así como la información que aporta cada elemento del mismo. (*The following graph describes a particular aspecto of a video game. Briefly describe what you think the graph represents as well as the information provided by each element in it*)



es un lugar con varias habitaciones, las conexiones que hay entre ellas y los elementos que hay en cada habitacion.

Cada una de las cajas representa una habitación y cada habitación tiene un nombre (playersOffice, Departament area, etc.) y una apariencia. las cajas de color azul son las 3 habitaciones principales desde las que se accede a las demas habitaciones las grises son las habitaciones desde las que se tiene acceso desde las principales y la habitacion en rojo (OfficeOnFire) es una habitacion en la que hay fuego. Cada uno de los elementos de color verde representa una puerta, el paso de una habitacion a otra y las flechas negras representan la direccion de ese paso. En cada habitacion estan señalados los elementos importantes que hay en ella (telephone, Extiguiseher, Alarm).

Según estos datos el juego podría tratar de averiguar el protocolo que hay que seguir para apagar un incendio: pulsar la alarma, avisar a los profesores para que desalojen el lugar, llamar por telefono, buscar el extintor y apagar el fuego.

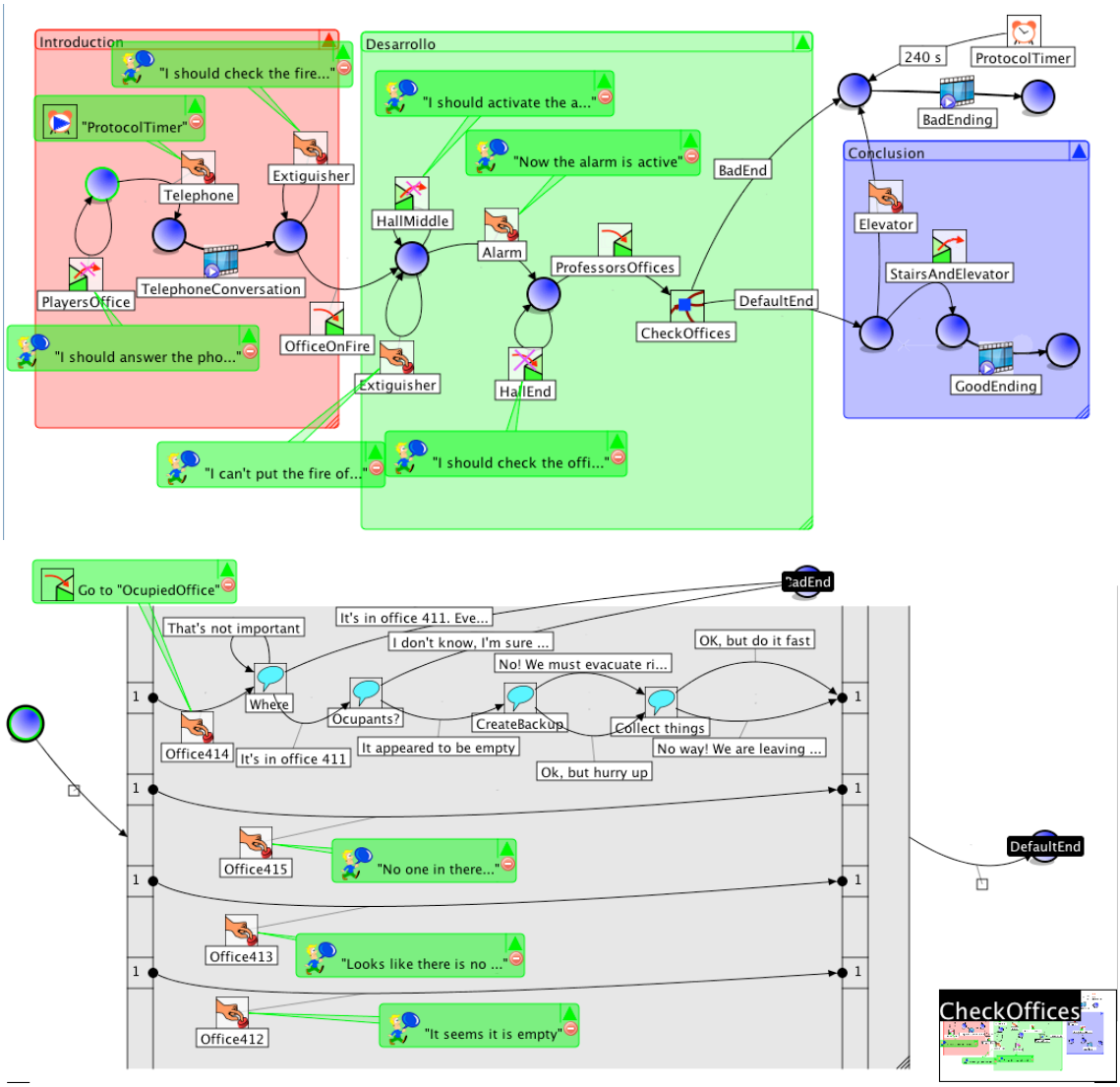
Este gráfico muestra la estructura básica del juego.

Se compone de los diferentes espacios en los que se desarrolla la historia(como el despacho, el hall, etc), las transiciones entre espacios(dejar oficina, salir del edificio, comprobar despacho, etc) y los objetos que se encuentra en cada espacio. Estos podrán ser parte de la escena los cuales el jugador no los podrá coger(alarma, despacho 412, 413, etc) o objetos que el jugador puede usar, coger o combinar con otros(teléfono).

Creo que cada nodo azul es una fase del juego y conectando las entradas y salidas de otros escenario por las puertas que dan acceso, tambien me parece que se indican los objetos

seleccionables del juego.
Describe como puede desarrollarse el juego. Con distintas escenas, algunas con objetos en ellas con los que puedes interaccionar (teléfono, extintor) y las posibilidades que tienes para ir a otras escenas según las acciones que hagas.
El gráfico representa escenas de un juego y las transiciones entre escenas. En las aristas aparece el nombre de la salida, y los cuadros amarillos representan zonas activas u objetos (mi respuesta se basa también en que conozco la lógica de e-Adventure y el juego en cuestión)
Este grafo corresponde a un videojuego en el que los escenarios son los rectangulos y supongo que su color dependerá del peligro de la escena. Las puertas son las acciones para pasar de una escena a otra. Y los rectangulos con borde amarillo corresponden a acciones sobre objetos.
La distribución de las habitaciones o escenas, como están conectadas entre sí y los elementos y zonas activas que contienen. Los rectángulos más grandes son las escenas, los bocadillos señalan objetos y zonas activas, y los cuadrados con puertas parecen representar las transiciones entre habitaciones.
Para mi el grafo representa lo que podría decirse el "mundo" del juego" , todos los elementos que componen ese mundo , y como se relacionan entre si los elementos. * Los rectángulos grandes representarían las zonas del mundo del juego , el entorno donde el jugador va a estar inmerso . * Los rectángulos grandes se relacionan entre ellos de manera que de una zona del mundo el jugador puede ir a otra . La direccion de la flecha indica de que zona a que zona el juegador puede pasar. * Los bocadillos representarían los elementos que están situados en cada zona del mundo . * Dentro de los bocadillos , el icono indicaria a que tipo de elemento se refiere Hay que tener en cuenta que yo tengo experiencia con con lo que para mi lo que este grafo representa es obvio , de todas maneras me parece una manera bastante intuitiva de representar un mundo en un juego de aventuras gráficas .
Parece que el objetivo del juego se trata en encontrar un incendio en una facultad. El usuario puede moverse por las distintas estancias (oficina, pasillo, escaleras y ascensor,...) y además en su oficina tiene acceso al teléfono.
Representa un mapa de localizaciones conectadas entre sí, al igual que elementos que se encuentran en las mismas y un atributo sobre el personaje en la escena. Las comunicaciones entre escenas y los tipos de objetos también están representados.

14. La siguiente figura (compuesta por los dos grafos, el de arriba y el de abajo) describe parte de un juego. Describe brevemente qué es lo que representa dicho gráfico así como la información que aporta uno de los elementos, según tu criterio. (*The following figure (including two separate graphs) describes part of a game. Briefly describe what you think each graph represents as well as the information provided by each element in the representation*)



El gráfico representa la historia del juego.Tiene tres partes la introducción, el desarrollo y la conclusión. En la introducción suena el teléfono y se comprueba que hay fuego en el edificio, en el desarrollo hay que activar la alarma y comprobar los despachos de los profesores. Checkoffice es un nodo que oculta parte de la historia, la cual se muestra en el gráfico de abajo. Por último la conclusión que puede ser buena o mala, dependiendo de las elecciones durante el juego.

El primer gráfico representa los estados por los que se pasa para terminar bien el juego, cada uno de estos estados está representado por un círculo azul. las flechas representan un paso a seguir y apuntan al estado en el que nos encontraríamos despues de realizar ese paso. Este gráfico tiene 3 partes:

Introduccion: en la habitacion "PlayersOffice" suena el teléfono, el usuario coge el teléfono y recibe el aviso de incendio, en este momento se inicia el temporizador para desalojar el lugar.

Desarrollo: El usuario pasa al Hall Middle, pulsa la larma y coge el extintor para apagar el fuego, pasa al ProfessorsOffice y avisa a todos los profesores para que desalojen el lugar.

Conclusion: Aqui hay dos opciones bajar en ascensor o por las escaleras solo se termina el juego con exito si se baja por las escaleras.

El segundo grafico representa lo que ocurre cuando se desaloja el Office de profesores, y las conversaciones que se desarrollarian. Los offices 415, 413 y 412 estan vacios. Al llegar al Office 414 se pueden elegir varias opciones y solo se llega al final con exito si se elige la opcion adecuada en cada momento.

Creo que grafo superior repredenta la logica de juego de cada escenario y el grafo inferior define las interacciones en el tiempo
De nuevo basándome en conocer el juego, la figura de arriba representa los cambios de estado posibles en el juego, con un icono en las aristas que identifica las acciones que disparan la transición y globos de texto con los otros efectos de la acción. La figura de abajo representa un subconjunto con ramificaciones, aparentemente es un "zoom" sobre uno de los estados vistos anteriormente (el que tiene un icono distinto). Viendo su estructura, parece como si se pudiese ir desde la izquierda hasta la derecha por cualquiera de los caminos (lo cual no es cierto).
El grafico superior representa mediante nodos el camino para llegar a una solución, que puede ser buena o mala. Cada cuadrado (con el dibujo) representa una acción, y las viñetas el resultado de esa acción. El gráfico de abajo es similar y representa la situación del juego dentro de 4 oficinas.
En el grafo de arriba vemos la trama de la historia. En una introducción estamos todavía en la oficina, y se ha respondido al teléfono, y se es informado del fuego, mientras que en el desarrollo se pulsa la alarma y se comprueban las oficinas, y a partir de ahí, dependiendo de las acciones del usuario tendremos un final feliz o un mal final. El inferior, parece ser las tramas que se seguirán al comprobar las oficinas, siendo en tres igual por que están vacías, y la primera dependiendo de como prosiga la conversación se hace un buen o un mal final...
En el primer gráfico, entiendo que cada uno de los circulitos con el degradado azul-blanco son "estados de juego". En conjunto forman una especie de autómata en las que se producen transiciones cuándo se cumplen determinadas condiciones. Cada transición parece estar asociada a un efecto. El segundo gráfico no lo tengo tan claro. Parece que todo parte de un estado inicial. En la parte de arriba está definida la estructura de una conversación, que se inicia a raíz de que se cumpla la condición o se ejecute algo de Office414. Las tres de abajo muestran también tres condiciones cumplidas y tres efectos. Y fuera un nodo de estado con "BadEnd" al que se llega desde la conversación. Y a la derecha, un "DefaultEnd" que imagino será un estado por defecto al que llegan todas las conversaciones si no se ha ido a "BadEnd".
Parece un mapa del juego, visto en lugares por los que hay que pasar, acciones que se pueden hacer en cada lugar y diálogos. Se muestra también a dónde conducen las localizaciones y lo que dice el personaje cuando efectúa una acción.
Primer grafo : El juego se representa mediante un grafo de estados , los jugadores dependiendo de las interacciones que tengan con el juego cambiaran de estado y se produciran distintos efectos * Los cuadrados de fondo representan una parte del juego , podria definirse como un capitulo o ni si quiera , simplemente una parte de la historia del juego * Los circulos son los estados del juego * Los estados del juego se relacionan por transiciones , cada transición se ejecuta cuando se produce un evento que es el que se describe en el icono asociado a la transición Los eventos a su vez pueden ejecutar efectos que son los que se indican con el bocadillo que lo acompaña. * Eventos : Los iconos con la mano y el boton : indica que el usuario ha interaccionado queriendo realizar dicha accion Los iconos con el cubo y la flechas de salida : indica que el usuario a realizado la accion de salir pero se puede o no se puede * Las transiciones en negrita , con el icono correspondiente , indican que se reproduce una slidescene o un dialogo y una transición automatica , sin que sea debido a una entrada . Como sugerencia personal lo enfatizaria cambiando de color la flecha de la transición o cambiando la forma del icono , para que se diferencie lo que es una transición realizada a partir de una entrada (caso de telephone) con lo que es lo que pasa en la misma transición (caso de telephone conversation) * Icono check offices : representa lo que sería una abstracción de alto nivel (representada a mas bajo nivel en el otro diagrama) de lo que puede pasar en ese estado del juego . Seria un estado con muchas relaciones a si mismo y a otros elementos .. el icono no queda claro si es estado , entrada o... :S * ProtocolTimer : el icono de protocol timer indica que el timer activado anteriormente en la

acción de ("Telephone") si llega a 240 segundos , este el juego en el estado en el que este , se realiza esa transición , del protocolo timer a el estado . Segundo grafo : Representa un estado , relacionándose con sigo mismo o con otros estados . Con una conversación , que dependiendo de las respuestas que se vayan obteniendo se producen transiciones al mismo u otros estados . Supongo que es una manera de realizar un grafo mas complejo en el que se incluye una conversación y mas transiciones. Los estados exteriores a la ventana gris serian los estados con los que se relaciona el estado a describir (la ventana gris) Los números de las barras laterales de la ventana gris indican el estado . Entre las dos barras se producen las transiciones , acciones efectos y el hilo de la conversacion de una manera similar a la del primer grafo.
Representa el guión del juego con los textos que el personaje dice según las distintas acciones que lleve a cabo.

15. Indica, de 1 a 5, la claridad de los dos gráficos anteriores (preguntas 12 y 13) para representar distintos aspectos del juego, dónde 1 significa "no queda claro en absoluto" y 5 significa "queda totalmente claro". (*Grade, from 1 to 5, the clarity of the information in the previous graphs, being 1 Totally nuclear and 5 Totally clear*)

¿Se entendía que el primer gráfico (pregunta X) representaba el mundo del juego (las escenas donde transcurre la acción y cómo se conectan unas con otras)? (*Was it clear that the first graph represented the world in the story?*)	4.3
¿Se entendía que el segundo gráfico (pregunta Y) representaba el flujo del juego (es decir, las acciones que el jugador deber realizar en base a las reglas del juego para poder progresar hasta cumplir el objetivo final)? (*Was it clear that the second graph represented the flow of the game?*)	3.5

16. Indica, de 1 a 5, la claridad de los elementos visuales que aparecen en el primer gráfico (pregunta 12) para representar distintos aspectos del juego, dónde 1 significa "no queda claro en absoluto" y 5 significa "queda totalmente claro" (*Grade, from 1 to 5, the clarity of the graphic elements used in the first graph to represent the different features of the game*)

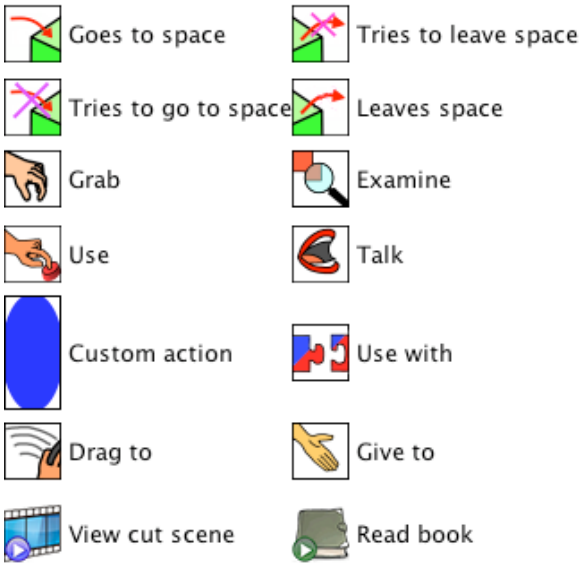
¿Se entiende que los rectángulos representan distintas escenas del juego? (*Is it clear that the rectangles represent different scenes in the game?*)	4.5
¿Se entiende que la caja que hay dentro de cada rectángulo sirve para definir el/los aspecto/s visual/es de la escena? (*Is it clear that the box inside the rectangle is used to define the visual appearance of the scene?*)	4.0
¿Se entiende que las flechas que unen los distintos rectángulos indican las conexiones entre escenas (es decir, que se puede ir de la escena de la que sale la flecha hasta la escena en la que termina la flecha)? (*Is it clear that the arrows that link the different rectangles represent the connections between scenes (i.e. the player can go from the scene where the arrow begins to the scene where the arrow end)?*)	4.9

17. Indica, de 1 a 5, la claridad de los elementos visuales del gráfico 2 (pregunta 13) para representar distintos aspectos del juego, dónde 1 significa "no queda claro en absoluto" y 5 significa "queda totalmente claro" (*Grade, from 1 to 5, the clarity of the graphic elements used in the second graph to represent the different aspects of the game*)

¿Se entendía que los "círculos" o "nodos" del grafo representaban puntos en el juego (es decir, distintos estados del juego)? (*Is it clear the "circles" or "nodes" in the graph represent points in the story of the game (i.e. different game states)?*)	4.3
¿Se entendía que las flechas representaban acciones que el jugador podía (o debía) realizar? (*Did you understand that the arrows represented the actions that the*	4.1

player could (or had to) perform?)	
¿Se entiendía que cuando hay una "escena de corte" (vídeo o diapositivas) el jugador no podía saltarla y debía verla obligatoriamente? (*Was it clear that when there is a "cut-scene" (video or slides) the player could not skip it and had to watch it compulsorily*)	3.0
¿Entendíste que la segunda parte de la figura correspondía a la representación del elemento "CheckOffices" de la primera parte? (*Did you understand that the second part of the figure represented the "CheckOffice" element in the first part?*)	3.0
¿Entendiste la función de los cuadros etiquetados como "Introducción", "Desarrollo", etc.? (*Did you understand the use of the boxes titled "Introduction", "core, etc.?*)	3.7
¿Se entendía que los "globos" que salían de las interaciones son los efectos que resultaban de realizarlas? (*Is it clear that the "balloons" that hanged from the interactions are the effects of performing them?*)	4.4

18. Sobre los icónos, esta es la guía de algunos de los iconos utilizados: (*About the icons, this is a key to some of the icons used by the system:*)



Se entienden todos (*All the icons are clear)*	**60%**	**6**
No se entendían: Los relacionados con space (*I had failed to understand the icons related to "space"*)	20%	2
Antes no había entendido el significado de los siguientes iconos: Custom action, drag to, use with (*I failed to understand the icons related to action, drag to and use with*)	10%	1
Antes no había entendido el significado de los siguientes iconos: Drag to (*I failed to understand the icons related to drag to*)	10%	1

2. Post-experience survey

This survey was presented to the users after they used the platform. It aimed to their perception of the platform and the representation elements used in it.

1. ¿Recomendarías utilizar <e-Adventure> WEEV cuando este terminado a alguien que quiera crear un juego educativo? (*Would you recommend using <e-Adventure> WEEV (ones it is finished) to someone interested in developing an educational game?*)

Sí (*Yes*)	**100%**	**9**
No (*No*)	0%	0

2. Aproximadamente, cuantos minutos tardaste en crear el juego de la experiencia guiada? (*Approximately how many minutes did creating the guided experience took?*)

Total (*Total*)	255
Media (*Mean*)	28

3. Si hiciste algo especial en la experiencia guiada, describelo brevemente a continuación. (*If you did something different in the guided experience, describe it briefly*)

Algunas correcciones que vi oportunas (como añadir un par de enlaces a un nodo virtual)
Ignoré el uso de Intro-Nudo-Desenlace (tampoco me quedaba muy claro cómo usarlo). Añadí algunos efectos adicionales.
Me costó meter el nodo random una vez montada la estructura inicial. Borré algunos nodos sin querer y tardé un poco en descubrir la manera correcta de hacerlo.
No hice nada especial

4. Aproximadamente, cuantos minutos tardaste en crear el juego de la experiencia no guiada? (*Approximately, how many minutes did you take to create the unguided experince?*)

Total (*Total*)	286
Media (*Mean*)	32

5. Si hiciste algo especial durante la experiencia no guiada, que no se ajustara a lo establecido inicialmente o que lo extendiera de alguna manera, descríbelo a continuación (*If you did something special during the unguided experience, beyond the recommendations, describe it briefly*)

Si el jugador compra un libro equivocado debe volver a la biblioteca a comprar otro libro.
Lo siento, no la he hecho
Me han faltado algunas conversaciones porque la aplicación no funcionaba bien.
No he hecho nada especial , he intentado seguir la historia que se pedia .
No me he salido de lo propuesto.

Propuse agregar al inventario el libro elegido y retirarlo al dárselo al jefe. Aunque no se si estará correcto.
Tuve que rehacer la estructura varias veces por no manejar correctamente las historias paralelas y optar por emplear la herramienta como un mapa de diseño más que para crear una aventura en eAdventure.

6. Evalúa la dificultad de las distintas parte del sistema <e-Adventure> WEEV, de muy difícil a muy fácil. (1-Muy difícil, 5-Muy fácil) (*Grade the difficulty of the different parts of the <e-Adventure> WEEV system, for very hard to very easy*). N=9

Question	Mean	Std.Dev.	Min.	Median	Max.
El proceso de crear juegos con <e-Adventure> WEEV es: (*The game creation process with <e-Adventure> WEEV is:*)	3.00	0.5	2	3	4
Entender los iconos de los botones utilizados por <e-Adventure> WEEV es: (*Understanding the icons used in the buttons in <e-Adventure> WEEV is:*)	3.44	0.73	2	4	4
Entender la representación y metáforas utilizadas por <e-Adventure> WEEV es: (*Understanding the representation and metaphor used by <e-Adventure> WEEV is:*)	3.11	0.78	2	3	4
Diseñar una historia atractiva para un juego con <e-Adventure> WEEV es: (*Designing an attractive story for a game with <e-Adventure> WEEV is:*)	3.44	0.73	2	4	4
Implementar la historia de un juego con <e-Adventure> WEEV es: (*Implementing the story of a game with <e-Adventure> WEEV is:*)	3.33	0.71	2	3	4
Utilizar <e-Adventure> WEEV para un desarrollador novel sería: (*Using <e-Adventure> WEEV for a novel developer would be:*)	2.67	1	1	3	4
Entender el tutorial es: (*Understanding the tutorial is:*)	3.78	0.44	3	4	4

7. Valoración general del sistema <e-Adventure> WEEV: Indica de 1 a 5 cuánto de acuerdo estás con las siguientes afirmaciones (puntúa desde totalmente en desacuerdo [1] hasta totalmente de acuerdo [5]). (*General valuation of the <e-Adventure> WEEV system. Use 1 for Strongly disagree and 5 for Strongly agree*)

Question	Mean	Std.Dev.	Min.	Median	Max.
Creo que <e-Adventure> WEEV es sencillo de usar (*I believe <e-Adventure> WEEV is easy to use*)	3.56	0.73	2	4	4
Creo que las metaforas y representaciones se entienden claramente (*I believe that the metaphor and representation are clear to understand*)	3.44	0.88	2	4	4

Creo que la representación gráfica permite entender bien la historia (*I believe that the representation allows a clear understanding of the story*)	3.89	0.93	2	4	5
Creo que los iconos utilizados se ajustan bien a lo que representan (*I believe that the icons used correctly fit what they represent*)	3.78	0.97	2	4	5
Creo que WEEV sería fácil de utilizar por un usuario novel (*I believe that a novel user would find WEEV easy to use*)	3.00	1.12	1	3	4
Creo que la nueva interfaz (botones, diálogos, etc.) es fácil de utilizar (*I believe that the new interface (buttons, dialogs, etc.) is easy to use*)	3.67	1.32	2	4	5
Creo que la nueva interfaz (botones, diálogos, etc.) se entiende bien (*I believe that the new interface (buttons, dialogs, etc.) is easily understood*)	3.44	1.13	2	3	5
Creo que la nueva interfaz (botones, diálogos, etc.) es agradable visualmente (*I believe the new interface (buttons, dialogs, etc.) is visually pleasant*)	3.56	1.33	1	4	5
Me gusta el nombre <e-Adventure> WEEV (Writing Environment for Educational Videogames) (*I like the name <e-Adventure> WEEV (Writing Environment for Educational Videogames)*)	4.33	0.50	4	4	5

8. Valoración del sistema (en relación con <e-Adventure>): Indica cuánto de acuerdo estás con las siguientes afirmaciones (puntúa desde totalmente en desacuerdo [1] hasta totalmente de acuerdo [5]). No necesitas responder a estas preguntas si no conoces <e-Adventure> en detalle. (*System valuation in relation to <e-Adventure>. Use 1 for Strongly disagree and 5 for Strongly agree. Do not answer these questions if you are not familiar with <e-Adventure>*)

Question	Mean	Std.Dev.	Min.	Median	Max.
Creo que <e-Adventure> WEEV es más sencillo de usar que <e-Adventure> (*I believe that <e-Aventure> WEEV is easier to use than <e-Adventure>*)	3.50	1.07	2	4	5
Creo que <e-Adventure> WEEV facilita la creación del mundo del juego (entorno navegable formado por escenas), en comparación con <e-Adventure> (*I believe that <e-Adventure> WEEV eases the creation of the world in games*)	3.88	1.25	2	4	5
Creo que <e-Adventure> WEEV facilita la creación de la historia del juego (flujo del juego), en comparación con <e-Adventure> (*I believe that <e-Adventure> WEEV eases the creation of the story of the games*)	4.50	0.53	4	4.5	5
Creo que <e-Adventure> WEEV puede sustituir completamente a <e-Adventure> (*I believe that <e-Adventure> WEE can completely substitute <e-Adventure>*)	2.25	1.04	1	2	4

Creo que integrar <e-Adventure> WEEV en <e-Adventure>, distribuyendo ambas herramientas de forma conjunta, sería positivo, pues facilitaría el desarrollo de los juegos (*I believe that integrating <e-Adventure> WEEV in the <e-Adventure> platform, creating a single distribution, would be positive because it will benefit game development*)	4.5	0.76	3	5	5

9. Indica qué partes de la representación te han parecido problemáticas (*Please choose the parts of the representation that gave you most problems*)

Multi-interacciones	78%	7
Nodos aleatorios	11%	1
Conversaciones	56%	5
Historias paralelas	22%	2
Otra: Ha dado algún problema al crear más de un actor personaje	11%	1
Otra: Funcionalidades que no quedan bien explicadas	11%	1

10. ¿Que cambiarías de <e-Adventure> WEEV? (*What would you change about <e-Adventure> WEEV?*)

El manejo de la herramienta a la hora de crear nodos o otros elementos es un poco complicado al principio, porque no sabes donde tienes que pincharle con el cursor para que se cree bien. Le tienes que dar a sitios específicos y eso es poco intuitivo cuando lo usas por primera vez. No se puede renombrar los linkspaces y la cutscene después de crearla. También al crear la cutscene si le das a ok sin pornerle un nombre no te la crea pero tampoco te avisa de lo que pasa. Estaría bien que mostrase algún mensaje. En lo de showText si escribes textos muy largos resulta luego complicado para borrarlo porque el botón de borrar está a la derecha y cuando vas a pulsarle se extiende el bocadillo y no puedes. Se podría poner mejor el botón a la izquierda.
- En la multiacción, en los nodos, todos los botoncitos (cambiar el número, eliminar, pinchar para añadir un arco) están demasiado juntos y en ocasiones pinchas en el que no querías
Agregaría opciones más complejas en un "submenú"
Algunos de los dibujos me parece poco representativo
Añadiría cosas, pero no creo que haya mucho por cambiar.
Añadiría más información sobre cómo funcionan las cosas (al estilo de la ayuda contextual de e-Adventure. Mejoraría algunas explicaciones que da el sistema y revisaría fuentes y tamaños de cuadros.
Cambiar lo que es cambiar , no cambiaria nada... Quizás en la parte donde se define la historia , los cuadrados traseros podrian ser directamente las escenas , creo que ayudaria a ver en que parte de el mundo (escena) se esta desarrollando la historia ... Cosas por pulir , al ser un prototipo , diría cosas como : * Las multi-interacciones ver como representarlas mejor . * El tema de las transiciones entre estados : diferenciaría las transiciones que se producen por una acción , de las que son "automáticas" como una cutscene .
Interfaz más amigable, quizás con ub botón de ayuda que indique que hace cada botón/comando.
Pondria los nombres debajo de los botones

11. ¿Que funcionalidades nuevas inclurías en <e-Adventure> WEEV, aparte de las que ya tiene? (*What new functionalities would you add to <e-Adventure> WEEV?*)

Incluiría más temas de ayudas, donde se explicase más para que sirve cada herramienta

Añadaría también efectos en los nodos, además de en las transiciones. Es decir, que cuándo se llegase a un estado, ya que puede llegarse por distintos caminos, pudieran ejecutarse una serie de efectos comunes a ese estado. - Que pudieran añadirse estados nuevos sin necesidad de hacerlo a través de arcos, crear nodos sin conexiones iniciales.
Agregar más opciones al nodo random.
Basicamente las que le faltarían para poder sustituir completamente a * Definición de los recursos que representan todos los elementos del mundo . * Definición de trayectorias en las escenas , posición inicial del personaje * Definición de animaciones * etc...
Cargar juegos ya hechos en e-Adventure, para permitir una edición de ida y vuelta. (alternativamente, preparar una distribución integrada en la que WeeV fuese la capa "superficial" y el editor de e-Adventure la parte de abajo.
funciones de copiar-pegar
Incluiria botones para añadir escenarios en la parte de la historia
Introduciría gestión de flujo de diálogos y también pondría mayor énfasis en representar las condiciones lógicas (si algo se cumple o no) También creo que podría ser beneficioso para dejarlo todo bien estructurado, establecer requisitos mínimos para el correcto funcionamiento. Por ejemplo, que el programa detectase, al emplear una transición de dar, si el objeto está en posesión del protagonista o si hay un diálogo, tener respuestas para cada pregunta o afirmación del personaje principal.
Un "wizard" que te explique paso a paso como crear un proyecto guiándote por el camino.

12. ¿Que te parece lo mejor de <e-Adventure> WEEV? (*What do you consider the best feature of <e-Adventure> WEEV?*)

Lo estructurado que está y las imagenes utilizadas que representan los objetos y las partes de la historia están bastante bien.
El aspecto visual! Sobre todo en World y Story. Comparado con como hay que hacer las cosas en e-Adventure esto es una gozada, poder hacerlo todo de manera gráfica, pinchando y arrastrando. Simplifica mucho el ver la historia de manera abstracta, sin tener que tratar con la representación gráfica en sí del juego.
Es más rápido para crear juegos que e-adventure.
La capacidad organizativa que otorga al diseñador.
La claridad con la que se puede ver el juego de un vistazo , tanto el mundo , los elementos que lo componen , como la historia. Ayuda al desarrollador del juego a tener una visión global del juego que está haciendo. Montaje rápido de escenas y la abstracción de los cambios de estado.
Muy sencillo crear una historia
Que está muy simplificado.
Se puede tener una vision global de un juego y tener claro desde el principio lo que quieres hacer de una manera muy sencilla

13. ¿Qué te parece lo peor de <e-Adventure> WEEV? (*What do you consider is the worst about <e-Adventure> WEEV?*)

Los fallos que tiene que he descrito anteriormente. Ademas, algunas opciones no funcionan del todo bien. El programa se queda colgado cuando creas un diálogo en el que hable otro personaje que no sea el principal.
Tras estar un rato con el programa abierto, todo empezaba a ralentizarse, sobre todo los TextField que, si escribías un poco rápido, no cogía todas las pulsaciones y se comía tildes y letras. - La distribución en

el apartado actors. Los elementos ocupan demasiado espacio y parecen poco organizados.
Algunos bugs (normal) y el no poder profundizar añadiendo elementos más concretos. Nada que no se vaya a mejorar invirtiendo en la herramienta.
El significado de los cuadros de intro-nudo-desenlace no era demasiado claro, las multiacciones tampoco las he entendido bien del todo (aunque eso es un problema de documentación más que de implementación). Las interfaces no me terminan de gustar, pero eso es cuestión de gustos. Sí las he percibido como funcionales y no he tenido problemas para usarlas en Mac (en windows no van demasiado bien).
Hay muy poca ayuda dentro de la aplicación.
La parte de la historia es un poco complicada para alguien que no sepa nada de videojuegos
Las multi-interacciones pueden ser complicadas de entender al principio.
Más complicado de usar que e-adventure.
Quizás se puede hacer un poco engorroso al edirar la historia , si esta es muy grande , el colocar los elementos de la historia para hacer hueco para nuevos elementos .. etc ..

14. En el sistema se va a añadir una parte de recomendaciones. Esto ayudará al creador del juego indicandole cosas como que debe poner más opciones en los nodos para incrementar el interés del juego o que debe poner evaluación en algunas partes. ¿Se te ocurre algun otro tipo de recomendaciones interesantes que se puedan hacer? (*A recommendation system is being planed for WEEV. This system will help the game creator with information about what could be done to improve the games. Can you think of any recommendation that would be interesting to make?*)

Recomendación de poner diálogos con personajes existentes que ayudarán a cumplir el objetivo.
Ahora mismo no, pero prometo pensar en ello...
En el momento en que estamos empleando una herramienta, estaría bien poder abrir un diálogo de requisitos para el correcto funcionamiento de la misma y alguna recomendación para que de un feedback completo al jugador.
No.
No...
Recomendaciones de no definir esa parte de la historia así sino hacerlo de otra manera mas sencilla y explicársela ... Recomendación de si algun elemento no lo ha usado en el juego (como podria ser un timer) decirselo que se puede usar y para que sirve...
Recomendar cuando se pueden utilizar nodos virtuales
Se le puede sugerir añadir más escenas o actores si tiene pocos para aumentar la interacción.
Sobre todo ampliar el menú con ayuda y más idiomas. Aunque sería muy útil hacer una parte avanzada dentro de la aplicación donde extendiese nuevas herramientas y destinada a usuarios más expertos.

15. Esta pregunta esta para que escribas cualquier otra sugerencia (opinion general, sugerencias, quejas, comentarios, etc.). La puedes dejar en blanco. (*Use this blank space to write any other suggestion you consider appropriate*)

En general está muy bien, se pueden crear juegos de una forma sencilla
Al añadir varios nodos de estados y volver a la parte de actores y añadir un nuevo actor, normalmente da una excepción y al volver al editor desaparecen los iconos y hay que reiniciar para poder utilizar la

herramienta de nuevo
Hay un error al darle al icono "add effect tool" y en la pestaña "Text" no me deja seleccionar "Speak NPC".
Me parece una idea muy útil y potente para organizar la historia, además de sencilla de utilizar, eso sí, sabiendo un poco como funcionan las cosas en e-Adventure.
Mi opinion personal es que me parece una idea muy buena , y con mucho sentido a la hora de representar un juego. También hay que tener en cuenta que para mi los diagramas de estado me son muy familiares debido a que estudio informática y no sé con que complicaciones podría encontrarse un usuario general. De todas me parece que esta genial! Enhorabuena!!
Por lo general, me ha gustado manejar la herramienta. Es cómoda y da una visión estructurada de lo que hay que hacer. Bastante buena para un diseñador. Algunas sugerencias que querría añadir son las siguientes: - Emplear "variables", de manera que simplifique el tener que coger el objeto adecuado en cada momento (esto viene, sobre todo, por lo que plantea a veces el emplear nodos aleatorios) Haciendo alusión a una variable en lugar de un objeto concreto, se puede facilitar mucho el diseño mediante WEEV. - Se debería poder "puentear" más fácilmente los nodos aleatorios, de forma que se pudiesen definir aleatoriedad entre más de dos opciones. - No se puede cambiar diálogos, a menos que se elimine y se vuelva a crear. - Se echa en falta poder emplear condicionales para simplificar la estructura. - Tener en cuenta la localización dónde estamos, al igual que lo que tenemos en el inventario. Si estamos en el sitio A y decimos "Salir de" no debería estar ninguna otra opción que no fuera A. Igualmente, en "Ir a", no deberían salir nodos no conectados a la localización actual en el grafo de localizaciones.
Problemas encontrados durante las pruebas: Si pongo una transición entre 2 escenas, el cuerpo me pide poner el segundo extremo de la flecha en el centro del cuadro. Si hago eso, en lugar de conectar las escenas, me ofrece la posibilidad de renombrar la apariencia. No termino de entender qué son los números en las multiacciones ni por qué los tengo que conectar como en el ejemplo. Al intentar exportar el juego el editor ha cascado y todo ha dejado de funcionar. En general, no he conseguido cargar lo generado con el editor. Después del cuelgue he abierto manualmente el editor y he intentado abrir el proyecto, pero no me carga ninguna escena ni ningún juego. A veces al intentar abrir el juego se abre una pequeña ventana que no hace nada. Hay que matar el hilo de Java para librarse de ella. Mi opinión general es buena. Para ser un producto distribuible le queda todavía trabajo, pero creo que las metáforas gráficas ayudan mucho y que son muy potentes. Son una contribución muy interesante de la que puede salir petróleo. Creo que tiene mucho valor dentro de e-Adventure y que podría ser nuestra "killer feature" si conseguimos definir un workflow que combine esto con lo que ya tenemos. Un trabajo estupendo, enhorabuena.
Un bug que he encontrado. Al querer editar un objeto desde la vista "Story" en vez de editar el seleccionado te edita el último que creaste.

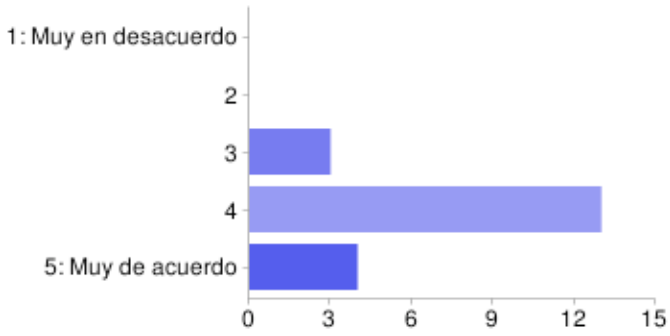
APPENDIX B: FULL SECOND FORMATIVE EVALUATION SURVEY RESULTS

The full results of the second formative evaluation are here included for completeness only. As the survey was made among Spanish users, it is here presented in its original Spanish.

1. Usability survey

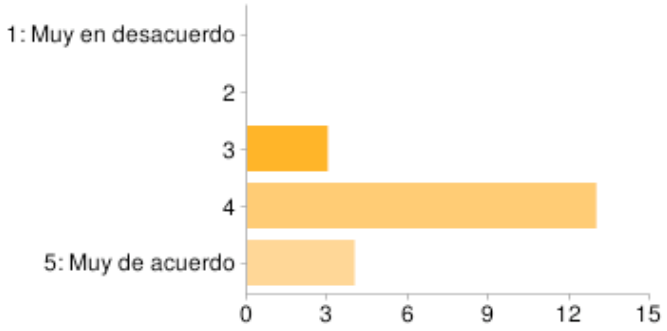
This survey is a direct translation of the one proposed by Lewis (Lewis, 1995) to measure system usability.

En general, estoy satisfecho con lo fácil que es usar WEEV (*Overall, I am satisfied with how easy it is to use WEEV*)



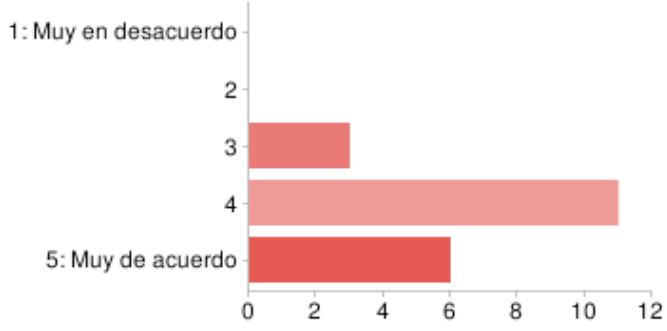
1: Muy en desacuerdo	0	0%
2	0	0%
3	3	15%
4	13	65%
5: Muy de acuerdo	4	20%

Fue sencillo utilizar WEEV (*It was simple to use WEEV*)



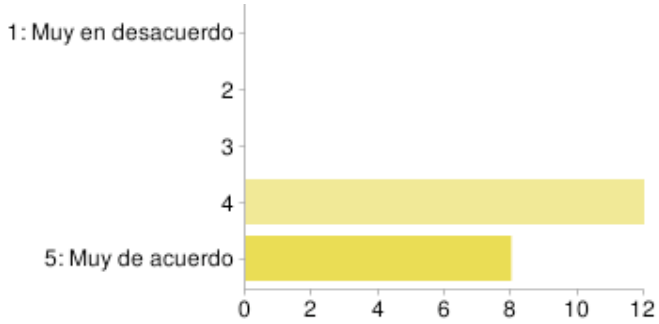
1: Muy en desacuerdo	0	0%
2	0	0%
3	3	15%
4	13	65%
5: Muy de acuerdo	4	20%

Puedo realizar juegos de forma efectiva usando WEEV (*I can effectively complete games using WEEV*)



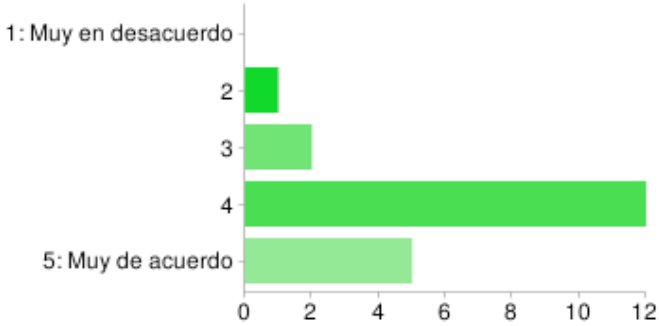
1: Muy en desacuerdo	0	0%
2	0	0%
3	3	15%
4	11	55%
5: Muy de acuerdo	6	30%

Soy capaz de crear juegos rápidamente usando WEEV (*I am able to complete games quickly using WEEV*)



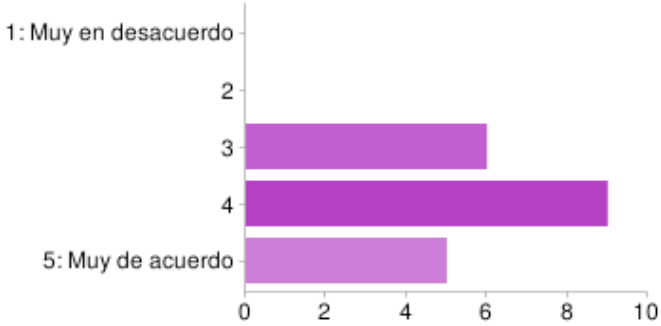
1: Muy en desacuerdo	0	0%
2	0	0%
3	0	0%
4	12	60%
5: Muy de acuerdo	8	40%

Soy capaz de crear juegos sin trabajo innecesario usando WEEV (*I am able to efficiently complete games using WEEV*)



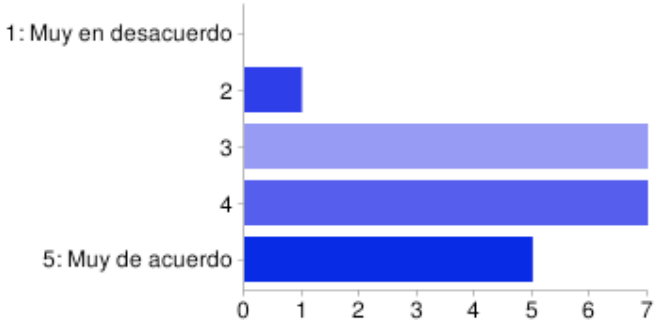
1: Muy en desacuerdo	0	0%
2	1	5%
3	2	10%
4	12	60%
5: Muy de acuerdo	5	25%

Me siento cómodo utilizando WEEV (*I feel comfortable using WEEV*)



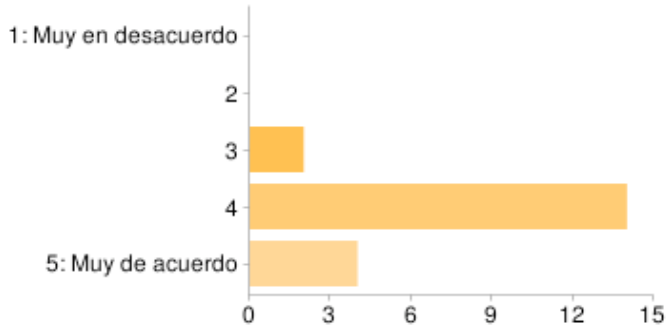
1: Muy en desacuerdo	0	0%
2	0	0%
3	6	30%
4	9	45%
5: Muy de acuerdo	5	25%

Fue fácil aprender a usar WEEW (*It was easy to leart to use WEEV*)



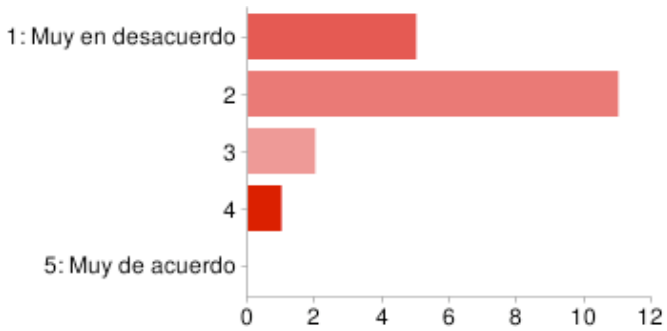
1: Muy en desacuerdo	0	0%
2	1	5%
3	7	35%
4	7	35%
5: Muy de acuerdo	5	25%

Creo que puedo ser productivo rápidamente usando WEEV (*I believe I became productive quickly using WEEV*)



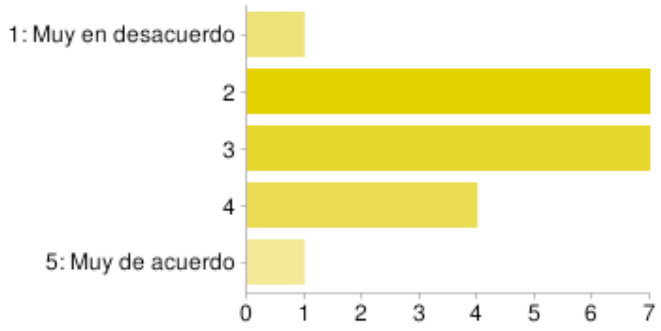
1: Muy en desacuerdo	0	0%
2	0	0%
3	2	10%
4	14	70%
5: Muy de acuerdo	4	20%

WEEV da mensajes de error que indican claramente como solucionar el problema (*WEEV gives error messages that clearly tell me how to fix problems*)



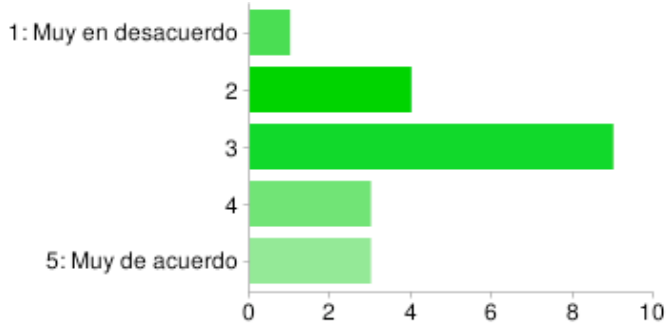
1: Muy en desacuerdo	5	25%
2	11	55%
3	2	10%
4	1	5%
5: Muy de acuerdo	0	0%

Cuando cometo un error con WEEV, puedo recuperarme fácil y rápidamente (*Whenever I make a mistake using WEEV, I recover easily and quickly*)



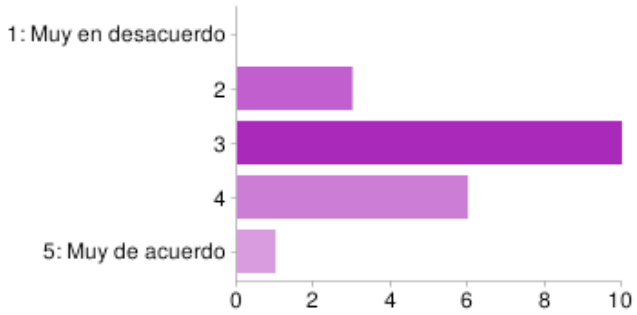
1: Muy en desacuerdo	1	5%
2	7	35%
3	7	35%
4	4	20%
5: Muy de acuerdo	1	5%

La información (como la ayuda, mensajes en pantalla y otra documentación) que da WEEV es clara (*The information (such as online help, on-screen messages, and other documentation) provided with WEEV is clear*)



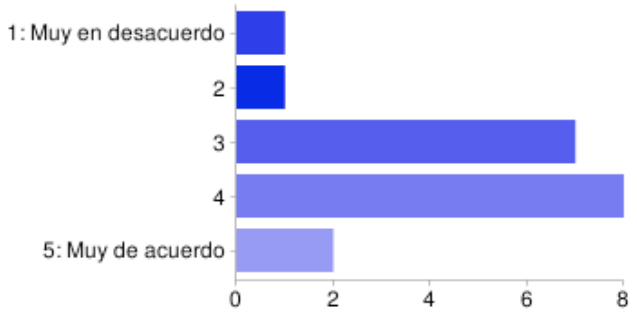
1: Muy en desacuerdo	1	5%
2	4	20%
3	9	45%
4	3	15%
5: Muy de acuerdo	3	15%

Es fácil encontrar la información que necesito (*It is easy to find the information I need*)



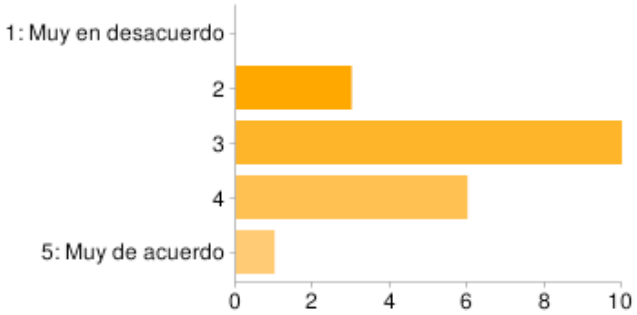
1: Muy en desacuerdo	0	0%
2	3	15%
3	10	50%
4	6	30%
5: Muy de acuerdo	1	5%

La información que da WEEV es fácil de entender (*The information provided for WEEV is easy to understand*)



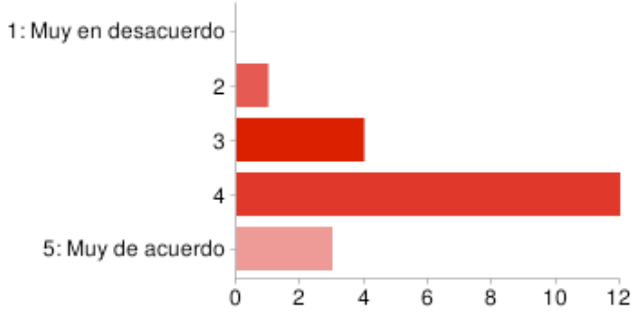
1: Muy en desacuerdo	1	5%
2	1	5%
3	7	35%
4	8	40%
5: Muy de acuerdo	2	10%

La información es efectiva para ayudarme a completar las tareas (*The information is effective in helping me complete the tasks and scenarios*)



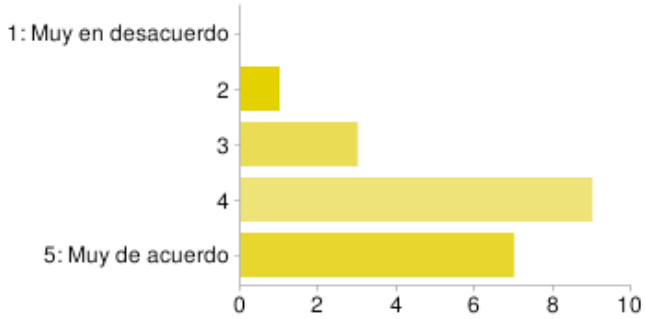
1: Muy en desacuerdo	0	0%
2	3	15%
3	10	50%
4	6	30%
5: Muy de acuerdo	1	5%

La organización de la información en pantalla de WEEV es clara (*The organization of information on WEEV screens is clear*)



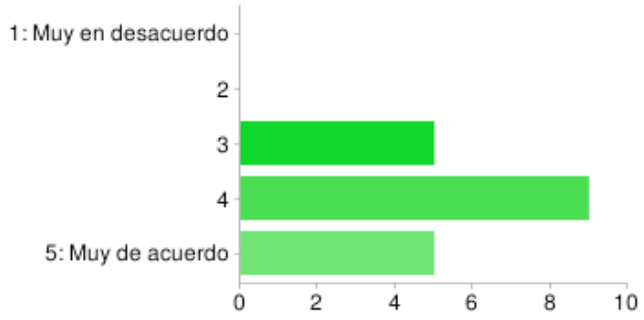
1: Muy en desacuerdo	0	0%
2	1	5%
3	4	20%
4	12	60%
5: Muy de acuerdo	3	15%

La interfaz de WEEV es agradable (*The interface of WEEV is pleasant*)



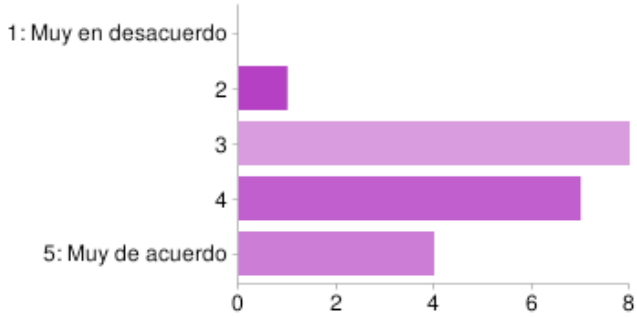
1: Muy en desacuerdo	0	0%
2	1	5%
3	3	15%
4	9	45%
5: Muy de acuerdo	7	35%

Me agrada utilizar la interfaz de WEEV (*I like using the interface of WEEV*)



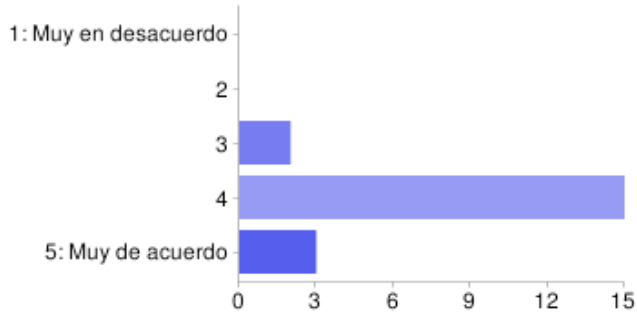
1: Muy en desacuerdo	0	0%
2	0	0%
3	5	25%
4	9	45%
5: Muy de acuerdo	5	25%

WEEV tiene todas las funciones y capacidades que espero que tenga (*WEEV has all the functions and capabilities I expect it to have*)



1: Muy en desacuerdo	0	0%
2	1	5%
3	8	40%
4	7	35%
5: Muy de acuerdo	4	20%

En general, estoy satisfecho con WEEV (*Overall, I am satisfied with WEEV*)

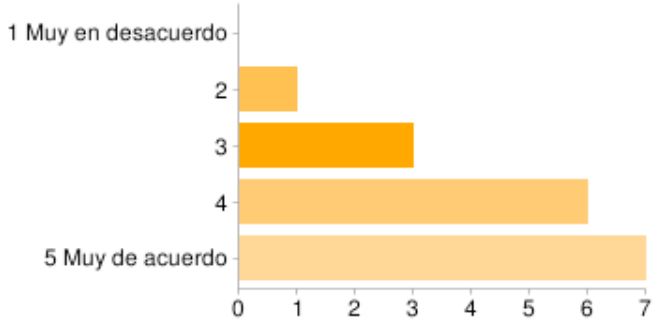


1: Muy en desacuerdo	0	0%
2	0	0%
3	2	10%
4	15	75%
5: Muy de acuerdo	3	15%

2. Perception survey

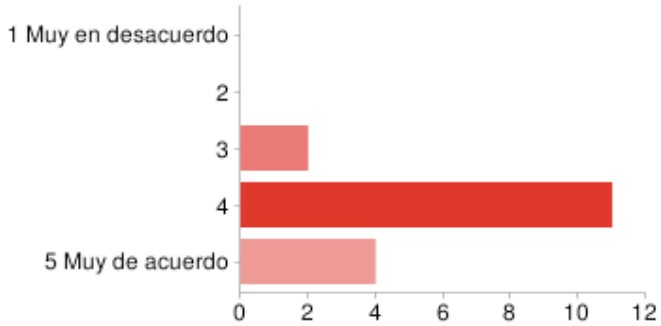
This survey was developed specifically to measure the perception of the different components of the system by actual users. It specially tried to establish if users familiar with both systems considered the WEEV metaphor and system as an improvement over the <e-Adventure> platform. A Likert 5-grade scale was used, with 1 being Strongly disagree and 5 being Strongly agree.

La metáfora (plantear los juegos como historias) se entiende bien (*The metaphor (games considered as stories) is clear to understand*)



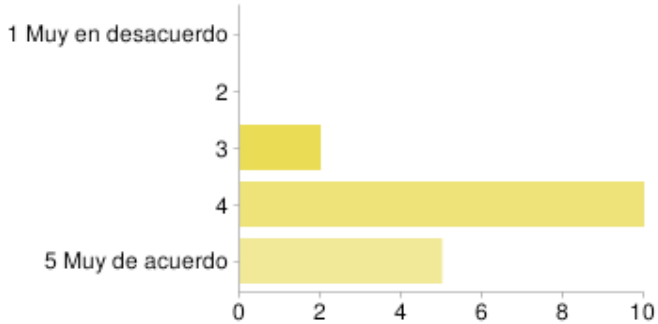
1 Muy en desacuerdo	0	0%
2	1	5%
3	3	15%
4	6	30%
5 Muy de acuerdo	7	35%

La representación (como se ve las cosas en pantalla) se entiende bien (*The representation (how things are displayed on the screen) is clear to understand*)



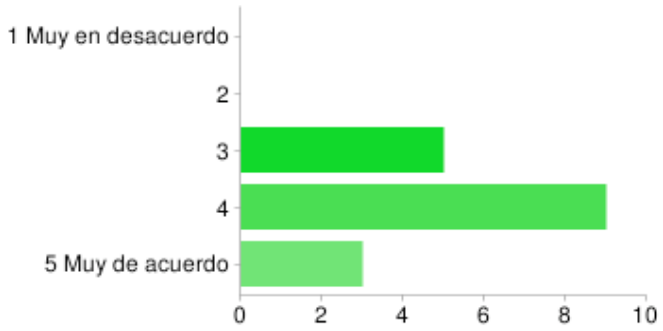
1 Muy en desacuerdo	0	0%
2	0	0%
3	2	10%
4	11	55%
5 Muy de acuerdo	4	20%

La representación de la historia se entiende bien (*The story representation is clear to understand*)



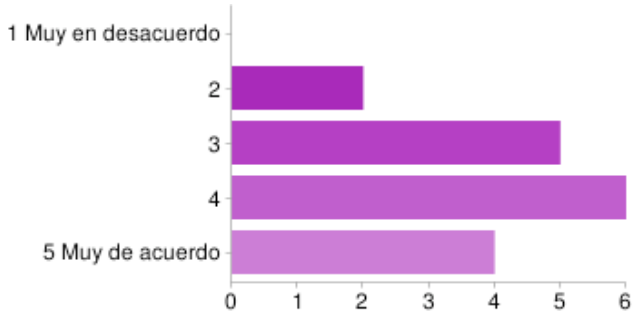
1 Muy en desacuerdo	0	0%
2	0	0%
3	2	10%
4	10	50%
5 Muy de acuerdo	5	25%

Los iconos se ajustan a lo que representan (*Icons are adequate for what they represent*)



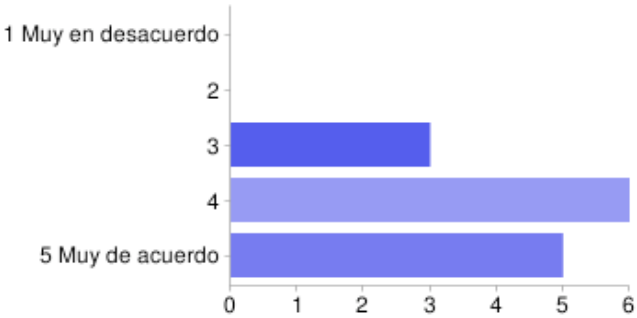
1 Muy en desacuerdo	0	0%
2	0	0%
3	5	25%
4	9	45%
5 Muy de acuerdo	3	15%

WEEV sería fácil de usar para un usuario novel (*A novel user would find WEEV easy to use*)



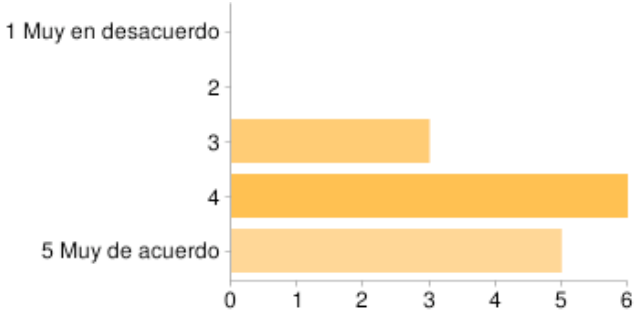
1 Muy en desacuerdo	0	0%
2	2	10%
3	5	25%
4	6	30%
5 Muy de acuerdo	4	20%

WEEV es más sencillo de usar que <e-Adventure> (*WEEV is easier to use than <e-Adventure>*)



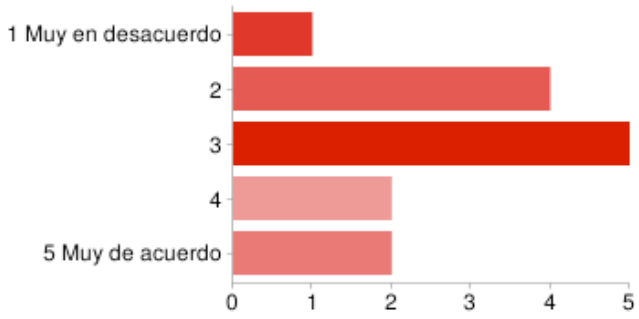
1 Muy en desacuerdo	0	0%
2	0	0%
3	3	15%
4	6	30%
5 Muy de acuerdo	5	25%

Crear historias con WEEV es más claro que con <e-Adventure> (*Creating stories with WEEV is clearer than with <e-Adventure>*)



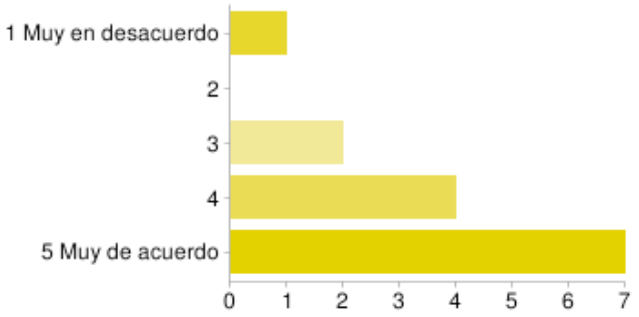
1 Muy en desacuerdo	0	0%
2	0	0%
3	3	15%
4	6	30%
5 Muy de acuerdo	5	25%

WEEV puede sustituir completamente a <e-Adventure> (*WEEV can completely replace <e-Adventure>*)



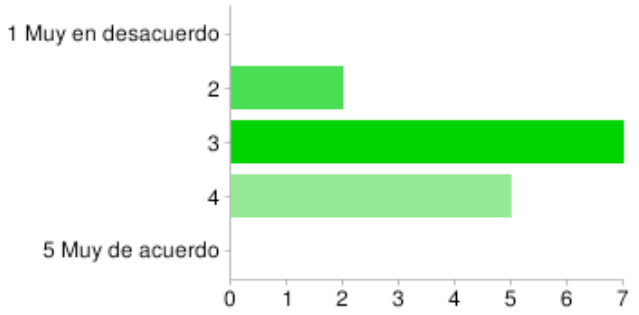
1 Muy en desacuerdo	1	5%
2	4	20%
3	5	25%
4	2	10%
5 Muy de acuerdo	2	10%

Integrar WEEV con <e-Adventure> facilitaría el desarrollo de juegos (*An integration of WEEV in the <e-Adventure> platform would greatly ease game development*)



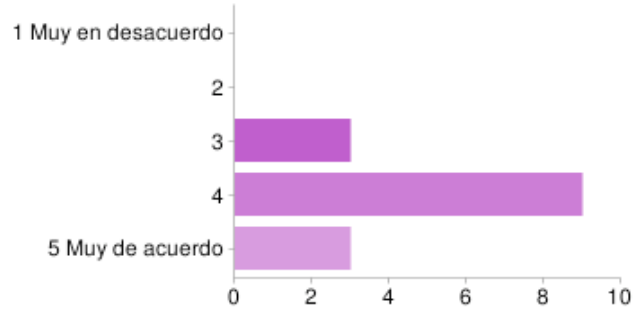
1 Muy en desacuerdo	1	5%
2	0	0%
3	2	10%
4	4	20%
5 Muy de acuerdo	7	35%

WEEV es muy útil para documentar un juego (*WEEV is useful to document a game*)



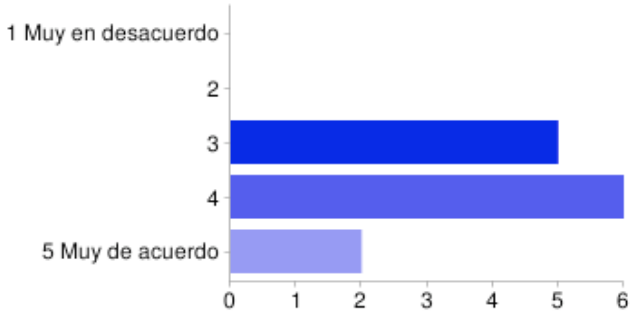
1 Muy en desacuerdo	0	0%
2	2	10%
3	7	35%
4	5	25%
5 Muy de acuerdo	0	0%

Añadir características educativas con WEEV es muy sencillo (*Adding educational features is very easy using WEEV*)



1 Muy en desacuerdo	0	0%
2	0	0%
3	3	15%
4	9	45%
5 Muy de acuerdo	3	15%

Añadir características educativas con WEEV es más sencillo que con <e-Adventure> (*Adding educational features with WEEV is easier than with <e-Adventure>*)



1 Muy en desacuerdo	0	0%
2	0	0%
3	5	25%
4	6	30%
5 Muy de acuerdo	2	10%

APPENDIX C: TUTORIAL FOR THE USE OF THE WEEV SYSTEM, A GUIDED EXPERIENCE

The following is a guided experience, similar to the one used in the evaluations of the system, in the creation of the salad game presented in the Uses Cases of this thesis (The salad game: A simple educational example).

1. Downloading and running WEEV

A beta version of WEEV (including all the necessary graphic resources for this experience) is available for download at sourceforge.net: http://sourceforge.net/projects/e-adventure/files/WEEV/WEEV_beta0.1.zip. The beta version is a zip file that you should unzip in the directory of your choice (e.g. "C:\WEEV").

The system is started running the bat file "Run e-adventure weev.bat" that you will find in the folder where you unzipped the beta version.

2. Follow the wizard

The wizard has several steps that you should necessarily follow. After each step, you should click the next button at the bottom to continue to the following step. Every step includes contextual help that is available by clicking the button with the question mark (?) at the bottom of the wizard.

Step 1: Welcome

In this step you should choose a name for your game. In this case you could call the game "SaladGame". You should also choose a working folder (the folder where the graphic resources will be placed during the session with the system).

Note: When you choose the working folder in the beta version, you will see several dialogs belonging to <e-Adventure> and possibly even an error message. Please ignore this and just click "OK" when prompted. This is because the underlying <e-Adventure> game is being created.

Step 2: Game type

The salad game is a "Photo-realistic interactive world". In reality, we will be creating the game using drawings, but the game type is still called this way. Do not choose "Adventure game" as no graphic resources for the main character are included, unless you can provide them yourself.

Step 3: Story structure

This is too simple a game as to require a story structure. It is recommended that you choose "No story structure". However, as this choice will not affect the workings of the game and is just used to organize the story, you can choose whichever structure you find more appropriate.

Step 4: Adaptation

We will not be using adaptation in this game, so choose the "No adaptation" option in this step. Were you to create a game that includes different

difficulty levels for example, you should choose the appropriate adaptation profile in this step.

Step 5: Actors

The game we are creating has four actors. You can add an actor by clicking its desired type (one of the tree buttons at the top of the wizard) and entering the name when prompted.

The actors are:

- Item: Salt.
- Item: Vinegar
- Item: Olive oil
- Part of scene: Salad

If you select an actor, more information is available. For example, you can write a description for it or add different "Appearances" that you will later be able to change among in the story. Besides, by clicking on the "looks" button when an actor is selected you can edit the graphic resources for it. Please use this button and define the different graphic resources for the items (salt, vinegar and olive oil). The necessary images are included in the beta version. After choosing the image for the object, click the "Create icon from image" button to make sure the player will be able to hold the item in the inventory.

Step 5: World

The world in the salad game is a simple world with just two spaces and links from each one to the other (Figure 77). These spaces are the kitchen (where the player will find the ingredients) and the dinning room (where the player will find the salad). The world panel starts out with one "Main space" and nothing else. Follow these steps to create the world needed for the salad game:

1. Change the name of the main space from "Main space" to "Dinning room" by right clicking on the space and selecting the "Change name" option.
2. Add a new space by selecting the appropriate tool (the "Add space tool") and then clicking on the panel. Name it "Kitchen" when prompted.
3. Set the resources. By clicking on a space and selecting the "Change appearance" option, you will be able to edit the graphic resources of the space. Select the appropriate image for each space for the ones included in the beta version.
4. Place the actors in the scenes. To do this select the appropriate tool (the "Add actor placement") tool and click on a space. The olive oil, vinegar and salt should go in the kitchen. The salad should be placed in the dinning room.
5. Place the actors within the scene. Once the actors are added to the scene and the scene has a graphic resource defined for it, you can

choose the exact place in the scene where the actor will appear. To do this, right-click the space and choose the "Place actors" option.

6. Select the "Add space link" tool from the top panel. You can now click on one space first (e.g. the kitchen) and then the other (e.g. the dinning room). You will be prompted for a name (e.g. "go to dinning room"). You should connect each space with the other.

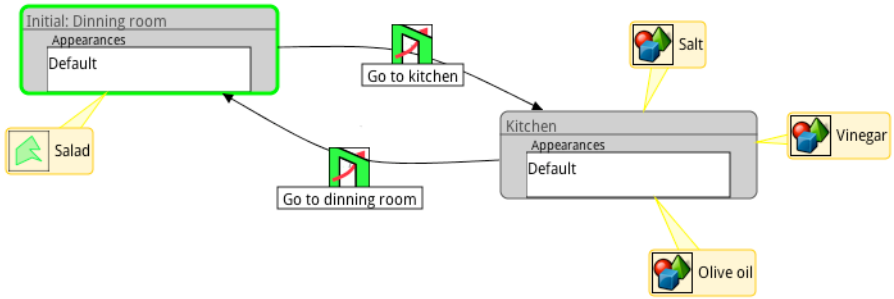


Figure 77 The world as created for the guided experience

3. Developing the story

Ones you finish editing the world and click next, a message saying "Finish with the wizard and start editing the story". You should click "Ok" in this message. This message indicates that once you start editing the story, you will no longer be able to go back in the wizard (you will still be able to modify the actors and the world, however).

The story we are creating is the following:

1. As the game starts, the player is at the dinning room. If he/she examines the salad, he/she will find out that it has not been dressed.
2. The player goes to the kitchen.
3. While in the kitchen the player grabs all three ingredients (olive oil, vinegar and salt). The player cannot go back to the dinning room without all three elements.
4. The player goes back to the dinning room.
5. The player must dress the salad using the ingredients in the correct order (vinegar or salt and then olive oil).

If the graphic resources where correctly assigned to the different elements in the previous step, the "Run" option in the "File" menu can be used at any time to try out the game. The "Convert" option can also be used to open the <e-Adventure> editor; any changes in the graphic resources of the game or the positioning of elements in the scenes done in the <e-Adventure> editor will be directly reflected in WEEV.

The tools and elements in the representation will not be detailed in this guided experience as they are described in the body of this thesis.

The story creation process can be gradual, starting by creating the main "branch" or the main elements or the story. The minimum story needed will include just the required actions of the player to successfully complete the game (Figure 78).

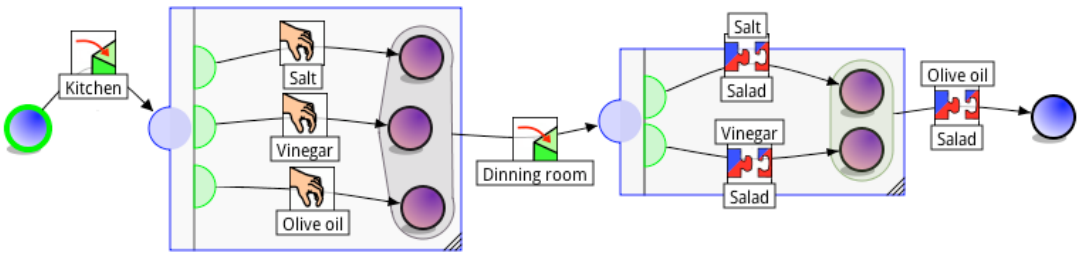


Figure 78 The main branch of the story in the guided experience

Once the main branch is created, the game must be made more interesting. Just by creating the main branch there is no room for the player to make mistakes, there is no feedback to find out more information, etc. More options and feedback can be added gradually. For example, the player will be forced to stay in the kitchen until all elements are grabbed, he will receive information when examining the salad, he will receive feedback when trying to dress the salad in an incorrect order, etc. (Figure 79)

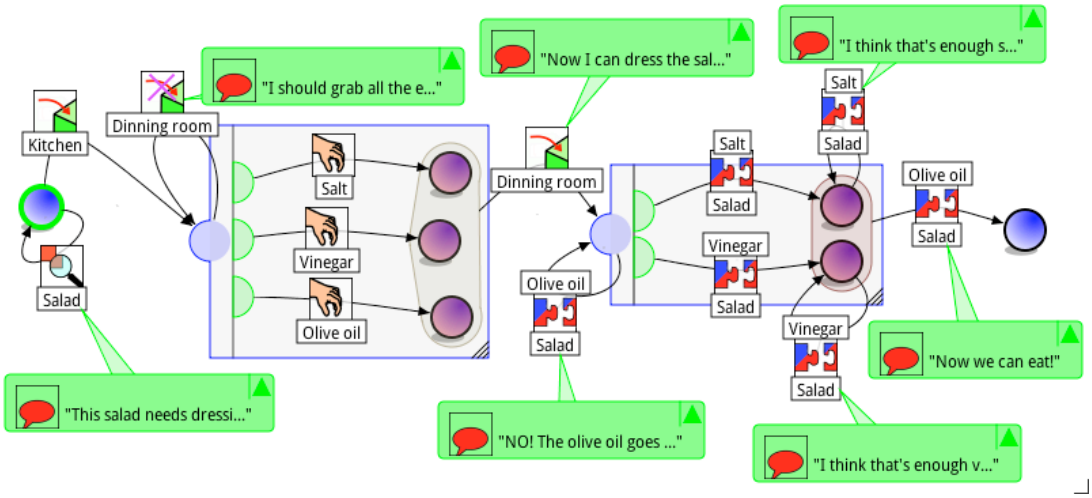


Figure 79 The story in the guided experience, with options and feedback

Lastly, educational features can be added to the story directly. In this case we choose to add evaluation to the game, giving different points (or taking them) as the user performs different actions (Figure 80).

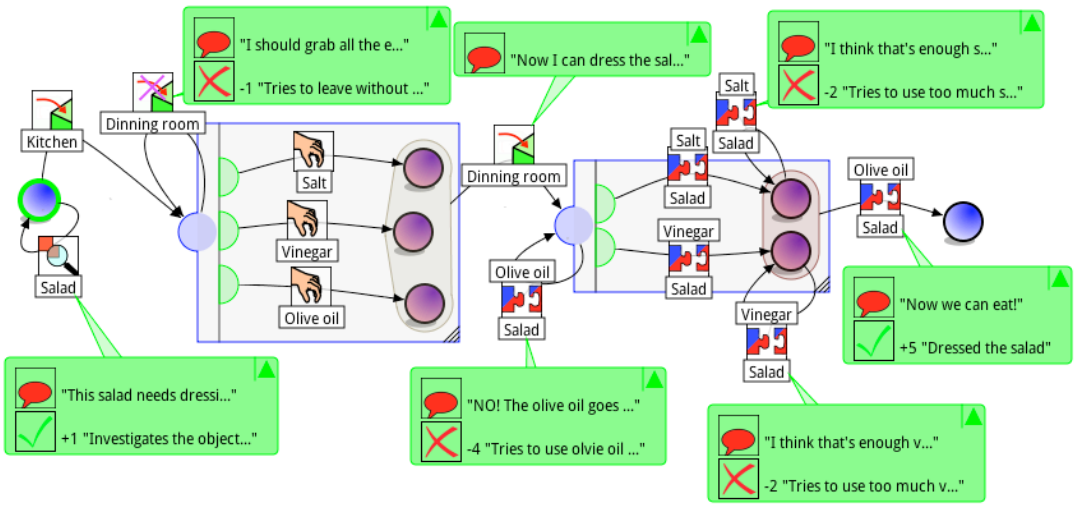


Figure 80 The guided expirence's story with evaluation

The game can be extended further, adding other possible mistakes (e.g. maybe a new ingredient?), providing more feedback, more evaluation or introducing hints.

This ends the guided experience. Remember that you can try out the game by choosing the “Run” option in the “File” menu.

APPENDIX D: RELEVANT PUBLICATIONS

This appendix includes relevant in regards to the subject of this work.

Marchiori, E., Torrente, J., del Blanco, Á., Moreno-Ger, P., Fernández-Manjón, B.: *A Visual Domain Specific Language for the Creation of Educational Video Games*. IEEE Learning Technology Newsletter, Vol. 12, No. 1, January 2010 (Special Issue on Game-Based Learning), pp. 36-39. 2010

A Visual Domain Specific Language for the Creation of Educational Video Games

Introduction

Educational video games and serious games are becoming more and more relevant as a complement to traditional instructional approaches. However, several barriers are in the way of the general adoption of this technology, such as the high cost or the integration of the games in the learning flow. At the <e-UCM> group at the Complutense University of Madrid we have developed the <e-Adventure>[1] platform for the creation of educational video games that addresses some of those problems. The current version of <e-Adventure> allows for the rapid creation of custom *point-and-click* adventure video games with low development costs [1].

In some cases, using COTS (*Commercial-Off-The-Self*) video games could solve at least partially some of these problems, but usually the available alternatives are very limited (e.g. using Civilization to teach History). When no COTS alternatives are found, a custom development is needed, but most educational professionals lack the necessary budget, tools and technical background. Using the <e-Adventure> platform allows educators to produce games without programming, but it is still perceived as too complex by many. According to our direct experience with educators at different levels, one of the most problematic issues is the difficulty to plan and develop a story using the system. In an effort to reduce this perceived complexity, we are creating a VDSL (Visual Domain Specific Language) to complement and enhance the creation of <e-Adventure> video games. This new approach provides a way to create games focusing first on the story behind them, which can potentially increase their educational value, as a strong narrative is one of the best game elements to support learning [2]. This story-based editor allows an educator to go from the game story flow to a working educational game without requiring technical knowledge.

Description of the Language

The new VDSL will represent the story using a graph-like structure, where the nodes represent different "points in the story" and the transitions indicate the flow. In video games, the flow of the story is driven by the interactions (i.e. actions in the game) of the user, and therefore they are represented as the transitions of the graph. The basic elements are based on the underlying <e-Adventure> model, but this approach could be applied to other tools and game genres. Many representation enhancements are used to reduce unnecessary complexity in the graph. The actions can also have consequences in the game world that do not alter the game flow and are added as properties of the graph (this includes mechanisms for tracking the performance of the students for later assessment).

This new system is created with the idea that a graphic representation, lacking some of the most complex elements of <e-Adventure>, can help teachers to acquire a better understanding of the games and increase their involvement in the development process. Applying a similar criterion, the new system is tailored into a "wizard", where all the basic information needed to create an educational video game is included so that novel users are guided though all the necessary steps.

[1] http://e-adventure.e-ucm.es

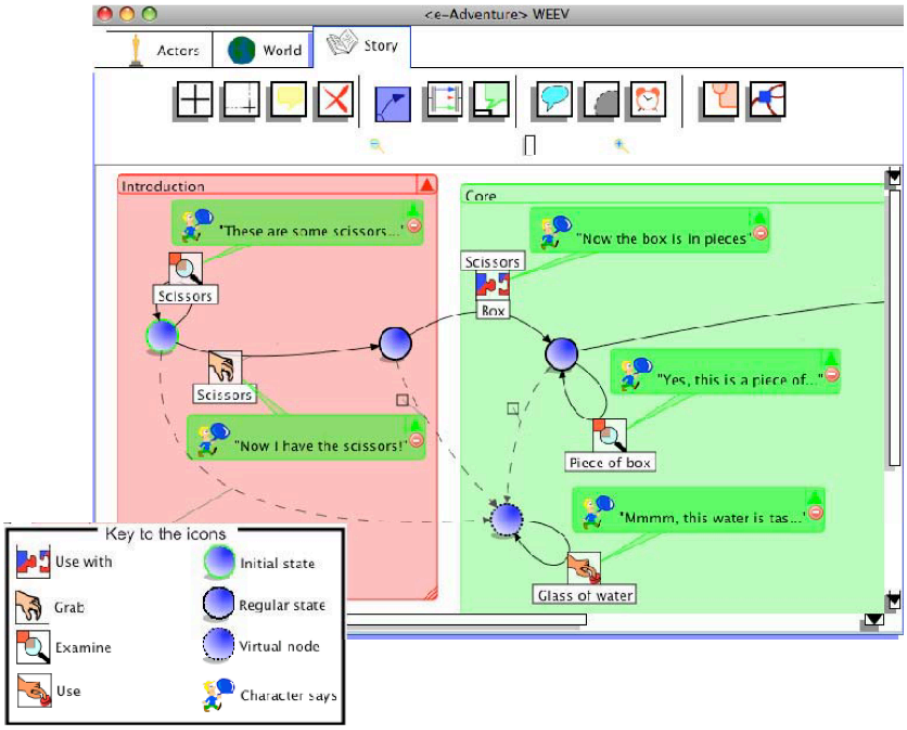


Figure 1: View of the story-flow editor, showing part of a game where the player has to grab some scissors and cut a box using them. The player can also choose to drink a glass of water at any time. Feedback is provided for the actions.

The game design process used in the wizard is based on research and real experiences on the field [3]. Besides, it borrows concepts from story writing; structural schemas will be used as a guide, to facilitate the development of the story in a meaningful way. Creating a good story is fundamental to achieve a high level of students' engagement and motivation. Even though we have no way to completely ensure a high quality of the story, this system will allow the authors to focus on its design by simplifying the rest of the development, which is a great advance.

Using a graphic representation has some other additional benefits over the traditional representation of the <e-Adventure> games. One of them is the possibility to generate recommendations for the user. These recommendations can cover the structure of the story (e.g. more o less branching, as needed) or its educational value (e.g. more assessments or more instructional content).

Besides, the new system will still have all the advantages found in the <e-Adventure> platform as the games created will be fully compatible and can be further edited using its advanced tools. These includes the possibility to export the game as Learning Objects [4] in compliance with the SCORM 1.2 or SCORM 2004 specifications.

Figure 2: Some steps of the wizard

Conclusions and Future Work

We expect that this system will simplify the development of custom educational games for novices in the field and allow developers to focus on the story. We intend to have a working version along 2010 and test it to determine if it really eases the development process in a controlled environment. After that, the new system will be included as part of the <e-Adventure> platform in future releases and distributed as open source software.

References

[1] P. Moreno-Ger, I. Martínez-Ortiz, J. L. Sierra, and B. Fernández-Manjón, "A Content-Centric Development Process Model.," *IEEE Computer,* vol. 41, pp. 24-30, 2008.

[2] A. Amory, "Building an Educational Adventure Game: Theory, Design and Lessons," *Journal of Interactive Learning Research,* vol. 12, pp. 249-263, 2001.

[3] M. D. Dickey, "Game Design Narrative for Learning: Appropriating Adventure Game Design Narrative Devices and Techniques for the Design of Interactive Learning Environments," *Educational Technology Research and Development,* vol. 54, pp. 245-263, 2006.

[4] J. Torrente, Moreno-Ger, P., Martínez-Ortiz, I., Fernández-Manjón, B., "Integration and Deployment of Educational Games in e-Learning Environments: The Learning Object Model Meets Educational Gaming," *Educational Technology & Society,* vol. 12, pp. 359–371, 2009.

Eugenio J. Marchiori
Complutense University of Madrid, Spain
emarchiori@fdi.ucm.es

Javier Torrente
Complutense University of Madrid, Spain
jtorrente@fdi.ucm.es

Ángel del Blanco
Complutense University of Madrid, Spain
angel.dba@fdi.ucm.es

Pablo Moreno-Ger
Complutense University of Madrid, Spain
balta@fdi.ucm.es

Baltasar Fernández-Manjón
Complutense University of Madrid, Spain
balta@fdi.ucm.es

This page is intentionally left blank

This page is intentionally left blank